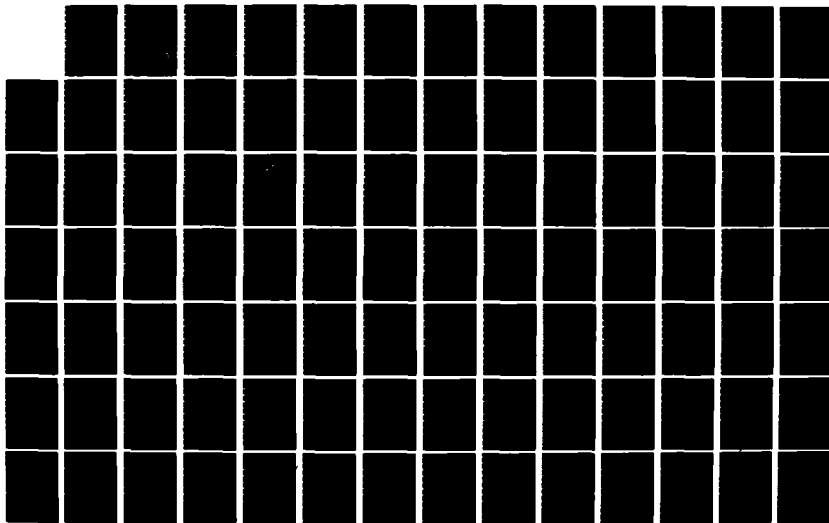AD-A138 503   GRID-BASED LINE DRAWING QUANTIZATION(U) AIR FORCE INST   1/2
              OF TECH WRIGHT-PATTERSON AFB OH SCHOOL OF ENGINEERING
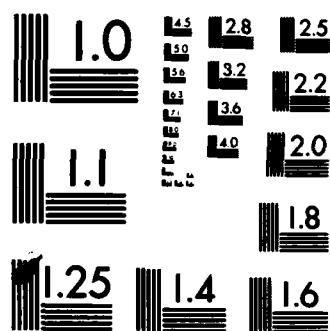   E R CHRISTENSEN DEC 83 AFIT/GE/EE/83D-16
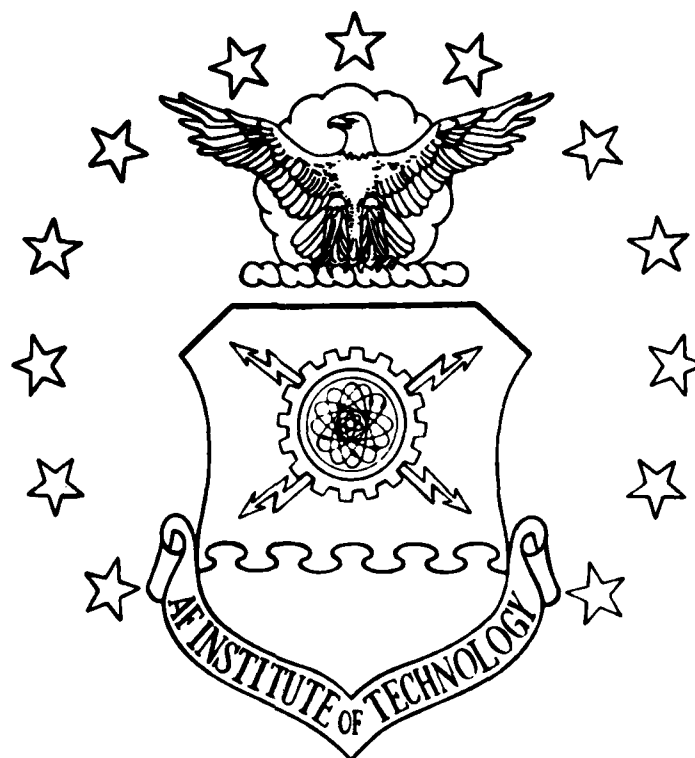
UNCLASSIFIED   F/G 9/2   NL

MICROCOPY RESOLUTION TEST CHART

NATIONAL BUREAU OF STANDARDS-1963-A

ADA138503

GRID-BASED LINE DRAWING

QUANTIZATION

THESIS

AFIT/GE/EE/83D-16   Eric R. Christensen
                     CPT,   USA

DTIC
ELECTE
S MAR 2 1984

A

DEPARTMENT OF THE AIR FORCE

AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

DTIC FILE COPY   84   02   29   05

GRID-BASED LINE DRAWING

QUANTIZATION

THESIS

AFIT/GE/EE/83D-16   Eric R. Christensen
                      CPT, USA

DTIC
ELECTE
MAR 2   1984

A

AFIT/GE/EE/83D-16

GRID-BASED LINE DRAWING

QUANTIZATION

THESIS

Presented to the Faculty of the School of Engineering

of the Air Force Institute of Technology

in Partial Fulfillment of the

Requirements for the Degree of

Master of Science

by

Eric R. Christensen

CPT, USA

Graduate Electrical Engineering

December 1983

Acknowledgements

I would like to thank my thesis advisor, Major Ken Castor of the Air Force Institute of Technology for proposing this thesis topic. I also would like to acknowledge that only through my faith in Jesus Christ as my Savior was I able to accomplish this project as Philipians 4:13 says "I can do all things through Christ which strengtheneth me". I am extremely grateful for all of the guidance I have recieved from Major Castor during the course of this project. I would like also to acknowledge my wife Georgetta for her understanding and support while I have been working on this thesis.

Contents

List of Figures

Abstract

This paper documents a quantitative analysis of the performance of the generalized chain codes when used to quantize sinusoids with specific periods and amplitudes. The analysis was performed using a software simulation of the various generalized chain codes using the triangular quantization scheme. The performance of the coding schemes was measured in terms of encoding rate and the area error in the quantization.

Comparisions were made of the performance of the codes as the amplitude to period ratio was changed. Comparisions were made when the same codes were used to quantize the same sinusoids. An attempt was made to quantize a rotated sinusoid, however the alogrithm implemented was to inefficient with respect to computer processing time. Finally the effect of a capture region of 1/2 of the grid size on the performance of the codes was examined.

Chapter I

Introduction

The ability to quantize, process, and store two-dimensional image data is extremely vital to many engineering and Department of Defense applications. Currently the most common method of quantizing two-dimensional image data is via two-dimensional sampling. To obtain high quality representations of the two-dimensional image a large number of samples must be taken, which requires a large amount of computer memory, and lengthy processing time. For many two-dimensional images two-dimensional sampling is extremely inefficient.

A class of two-dimensional images for which two-dimensional sampling is extremely inefficient is line drawings. A line drawing is a two-dimensional image consisting totally of thin lines drawn on a contrasting background (Ref 1:237). The line drawings may then be classified as "regular" if they are composed of well defined geometric figures with relatively few arcs and straight lines such as alphanumeric text or engineering drawings (Ref 12:1). The line drawings are classified as "free-form" if it represents terrain contours, geographical maps, or other natural-object boundaries (Ref 12:1). Using the current techniques of two-dimensional sampling, in terms of computer data a geographical map will typically contain 10-40 megabytes

of data (Ref 18:619). The data generated by the two-dimensional sampling contains a lot of redundant data and a need for more efficient means of quantizing and encoding was realized.

A family of quantization and encoding schemes has been developed which yields a more efficient method of quantizing and processing two-dimensional line drawings. This family of schemes is commonly referred to as the generalized chain codes. Presently there does not exist a specific method of analyzing the performance of the generalized chain codes in a quantitative manner as all current measures are primarily qualitative in nature.

The objective of this thesis was to continue the effort begun by Castor and Neuhoff and continued by Jones in developing a quantitative comparision of the performance of various forms of the generalized chain codes when used to quantize certain periodic waveforms (Refs 1,10,19). This thesis explored the performance of the various chain codes in quantizing sine waves possesing different spectral characteristics and the effect of quantizing in terms of the first grid intersect with in one half of a grid size of the ring versus the first grid intersect on the ring. Additionally work was begun on the problem of quantizing sine waves rotated at some positive angle above the X axis. The performance measures being used for the analysis are code rate and accuracy of the of the various codes when they are used

I-2

to quantize the various line drawings. The accuracy
performance measure is obtained by determining the total area
lying between the original line drawing and its digitized
approximation and normalizing the total area per unit length
of the orginal line drawing. The code rate performance
measure is obtained by determining the number of bits required
to encode a specific line drawing and normalizing the number
of bits required per unit length of the original line drawing.

The assumptions made in this thesis are the following:

1. The line drawings will not cross over themselves.

2. The grid is located in the right half of a plane of a
    cartesian coordinate system such that the vertical grid
    lines are parallel to the Y-axis.

3. The line drawing can be defined by a function which is
    at least piece-wise continuous and invertible, and
    whose domain extends from zero to some finite positive
    number. (Note: The function does not have to be
    invertible in the strict mathematical sense, but its
    inverse must be implementable in the software.)

4. The grid size of the quantizing gird is small enough
    such that the derivative of the function defining the
    line drawing has at most one zero between any two
    vertical grid lines.

These assumptions are made to limit the scope of this thesis
and to aid in the synthesis of the line drawings. The defining
of the line drawings by some function will aid in the
collection of data since line drawings with the same specific
characteristics can be synthesized repeatedly for quantization
by the different quantization schemes. Additionally by
defining the line drawing by some function the error

I-3

statistics will be easier to evaluate since the statistics for the original line drawing will remain the same for each generation of the line and only the measurements of the quantized line will change.

The method of presenting the material in this thesis closely parallels the approach to the problem. Chapter II is a review of the current literature pertaining to the subject and presents the background material necessary for understanding the subject. It presents an overview of what line drawings are, a general discussion of the grid intersect code and the generalized chain codes, the triangular and parallel quantization schemes, an overview of suggested performance measures for generalized chain codes, and a description of the performance measures to be used in this thesis to evaluate the performance of the generalized chain codes. Chapter III is a brief overview of the algorithm developed by Jones used to synthesize line drawings and quantize line drawings using the triangular quantization scheme (Ref 10). Chapter IV presents the modifications made to the Jones' algorithm. These modifications were necessary to analyze the performance of the chain codes to quantize the sine waves rotated at an angle above the X-axis, and to allow intersection points lying within 1/2 of a grid size to be considered on the ring for quantization purposes. Chapter V presents the analysis of the performance of the generalized chain codes with regard to spectral characteristics of the

line drawing being quantized using the triangular quantization scheme. This analysis will also consider the two interpetations of defining the next node to be quantized. Chapter VI presents the conclusions and recommendations for further study.

Chapter II

Background

A line drawing is one of man's most common means of communication, and its processing by computer has attracted the attention of computer engineers for more than a decade (Ref 3:57). A line drawing is a two dimensional (2d) image where information is conveyed by the shape, size, manner of interconnection, and location of thin lines on a contrasting background (Ref 1:237). Examples of such drawings include charts, graphs, maps, and printed or handwritten alpha-numeric text (Ref 1:237). In these types of drawings the thickness of lines, their color, or texture of the background have little or at most symbolic significance (Ref 3:57). For many years there has been an awareness that much benefit could be gained by replacing the traditional paper medium of storage for line drawings with a computer readable medium (Ref 8:619).

A line drawing is a form of communication, and when we speak of processing line drawings, we are in reality referring to processing information (Ref 11:31). The phrase "processing line drawings" is at times used to refer to two different operations. one is the extracting of 2d line structures from line drawing images and the other is processing of line structures to extract information of interest (Ref 3:60). This thesis will be concerned with the second-the processing of line structures as information. Three terms which will be used frequently are "images", "line drawing", and "line

structures". These terms are defined by Freeman as follows:
An image is a natural visual object that is characterized by
the 2d spatial variation of brightness or color or both. It
is objectively sensed by the normal human eye and does not
depend upon assignment of meaning for definition. Examples
are paintings, photographs, and printed text. A line
structure is a geometrically defined concept, consisting of an
assembly of points, line segements, and curve segments in
Euclidean space. This assembly need not be connected. Line
structures can be precisely defined in an appropriately
selected coordinate system. The term line drawing is used to
denote an image used to convey information about a 2d line
structure (Ref 3: 57-60).

A line sturcture can be represented visually by an image.
the use of an image to convey information is, as in all forms
of modelling, a highly subjective process, dependent upon the
observer, his past experience, the time, the place, and the
context in which the information is conveyed (Ref 3:58). Thus
a line drawing is first of all an image--it can become a line
drawing in the mind of the viewer if he so perceives it (Ref
3:60).

A line structure can originate in one of three ways, and
its origin significantly affects the way which it can be
processed (Ref 3:60). First, a line structure may be
abstractly specified in a geometric sense. This is the case
where line drawings serve as a model for line structures (Ref
3:60). A second source of line structures if referred to as

"tracing". Tracings are derived from the physical world, and if planar, can be reperesented by line drawings (Ref 3:60). The third source of 2d line structures is images. In this case, the line structure serves as a model for the information conveyed by the image (Ref 3:60).

Line structure processing problems can be grouped into four categories based upon input and output operations. These four groups are the following: 1. Analysis; 2. Synthesis; 3. Manipulation; 4. Pattern Recognition (Ref 3:60). In analysis the line structure is given and its characteristics are determind. The synthesis of line structures is their generation with certain characteristics and their display by means of labeled line drawings. Manipulation encompasses the problems in which line drawings are subjected to different transformations. Pattern Recognition involves the classification of line structures (Ref 3:60).

The requirement to have ready and rapid access to vast amounts of 2d image data has led to a requirement to be able to efficiently quantize, process, and store such data. One of the common methods of quantizing 2d images is through 2d sampling (Ref 10:1). The process involves large amounts of ⌐er memory and time.

⌐lass of 2d images in which rapid access is needed in m ⌐ pplications is maps. The ability to store maps in a computer data base versus using a paper medium of storage would significantly decrease the time required to access a

specific map. Using current quantization techniques the amount of data required to encode a single map for computer storge is in the range of 10-40 megabytes (Ref 8:619). This data by nature of the quantization method used contains much redundant information (Ref 8:619).

An efficient method of quantizing and encoding line drawings method uses a superimposed cartesian grid on the line drawing to be quantized (Ref 9:5). This method is commonly called the grid - intersect quantization method, and line drawings that are quantized using the method are referred to as grid-based line drawings (Ref 9:5). The horizontal and vertical grid line intersections are referred to as nodes. The quantization of the line drawing is accomplished by selecting the node that is nearest to the intersection of the line drawing and the grid lines (Ref 9:5). If the line drawing crosses the grid lines in such a manner that a single node would be selected twice in succession it is only selected once for encoding (Ref 9:5). When successive nodes are connected by straight line segments a chain is formed (Ref 9:8). The nodes are encoded by a method in which each node is assigned a binary number which represents its relative position to the previously selected node (Ref 10:3). The sequence of quantizing and encoding the line drawing is referred to as chain coding.

THE GRID INTERSECT CODE

The simplest chain code is the grid intersect code or (8 - point) chain code which forms the basis for the family of

generalized chain codes (Ref 10:3). This code allows selection of one of eight possible next nodes. These nodes form a square which is referred to as a ring. Figure II-1 shows a typical grid and the current node with the eight possible next nodes.
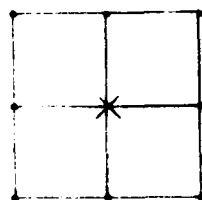
x - current node

. - next possible node

Figure II-1 Possible Next Nodes

Letting the grid mesh be of size q the next node to be selected must lie within a distance of either q or $2\sqrt{q}$ from the current node (Ref 6:1). When all possible nodes lie on a boundary of a square of side 2q and centered about some point A, the square boundary is called a ring 1 (Ref 6:1). Figure II-2 shows a ring 1 code grid.

q

q
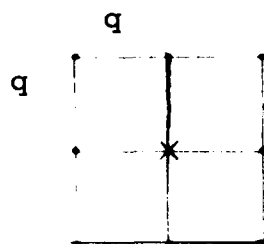
X- Center of grid
.- Possible nodes
q - Grid mesh size

Figure II-2 Ring 1 code grid

The encoding procedure consists of locating the first line drawing and grid line intersection on ring 1 and the selection of the grid node nearest to the intersection as the next point of the quantized set (Ref 10:4). Figure II-3 is an example showing the nodes that would be encoded for the line drawing shown
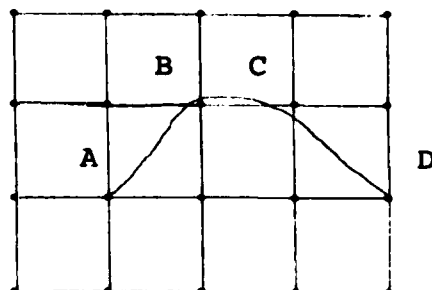

Figure II-3   Example Illustrating Grid Intersect Code
Point A was the last previous node to be encoded,
Points B,C,D are the newly encoded nodes
in alphabetical order.

A binary encoding procedure is to associate an octal integer (3 bit binary number) with each node of the ring as shown in Figure II-4 (Ref 10:5).
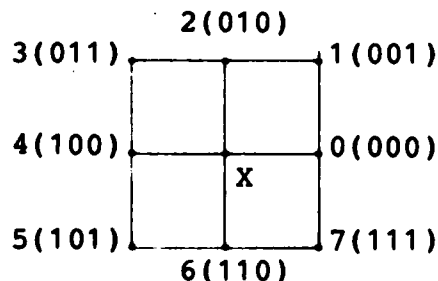

Figure II-4 Encoding Assignments For Ring 1 Code

The assignment of the octal integers to each node is arbritary but must be consistent for encoding and decoding. The grid nodes of the quantization process are encoded by representing the relative position between nodes by one of the eight octal integers. For example the line drawing of Figure II-3 would be represented by the chain 107.

HIGHER ORDER CODES

The quantization and encoding procedure can easily be extended to higher order codes. This is done by allowing the set of possible next nodes to include nodes from rings other than ring 1. The generalized chain code can include 16, 24, 32, 48 or even more permissible next nodes (Ref 6:1)

The generalized chain codes are defined in terms of rings contained in the code (Ref 7:1). The ring is defined to be a square of side 2nq with 8n equi-spaced nodes on the perimeter, located such that nodes are located at each corner of the square(Ref 7:1). A vector directed from the center of the ring to any node on the ring is called a "link". (Ref 12:10) A chain code is formed by selecting a combination of rings. The code is identified by the set of rings which it contains. The grid intersect code (8-point) chain code is referred to as the (1) code, and the code consisting of rings 1 and 2 as the (1,2) code. The set of permissible next nodes for the (1) code are as shown in Figure II-1.The set of permissible next nodes for the (1,2) code are as shown in Figure II-5.
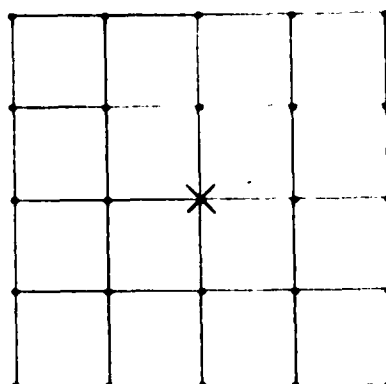
X-Current Node

.-Permissible next Nodes

Figure II-5 Permissible Next Nodes (1,2) Code

The quantization procedure for the generalized chain code is more complex than that used for the (1) code. Through the dedfinition of sets of "link gates" many different methods of selecting links that best approximate the curve can be chosen. There are also many different configurations possible for the definition of these link gate sets (LGS)(Ref 12:13). Each particular definition and configuration yeilds a distinct quantization scheme. Two of the more common schemes are the triangular and parallel schemes. These two schemes will be the only ones discussed in this paper.
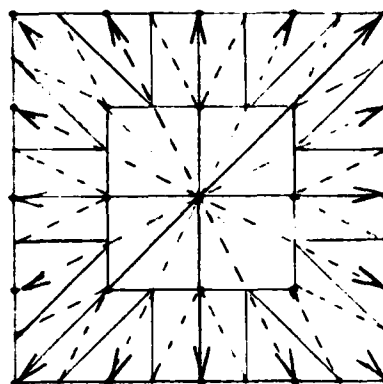
Parallel Quantizing Scheme

The LGS of the parallel quantizing scheme are defined by choosing the highest-order-ring contained in the code and finding the midpoints of all pairs of adjacent nodes on that ring. The next step is to draw two parallel lines for each possible link of the ring from the two neighboring midpoints on the sides of the link. The two parallel lines are called

midpoint lines. The LGS is thereby defined as the parallel ring segments cut from each ring the the adjacent midpoint lines. This process is repeated for each ring contained in the code to define the LGS for each level of the code. The set of all possible LGS for a particular code is called the "template" for that code (Ref 12:15).

The actual quantization procedure now consists of a search for the highest-order-ring link for which all link gates intersect the curve. The search begins with the center of the template lying on the last encoded node. The next step is to locate all intersections points of the curve and all rings contained in the code. Then a check is made to determine if all intersections are contained in the LGS for the highest-order-ring of the code if so then the associated node is selected for encoding. If all of the curve and ring intersections are not contained in the LGS for the highest-order-ring of the code then the process is repeated for the next lower order ring of the code until all curve and ring intersections are contained in the LGS of a ring. The LGS of the lowest-order-ring will always meet the quantization criterion (Ref 12:15). This process is repeated until the line drawing is quantized and encoded. Figure II-6 shows the templates associated with Rings 1,2, and 3 using the parallel quantization scheme. Figure II-7 shows the generalized form of a LGS for the parallel quantizing scheme.
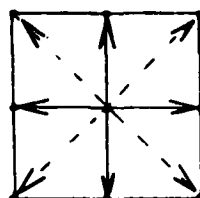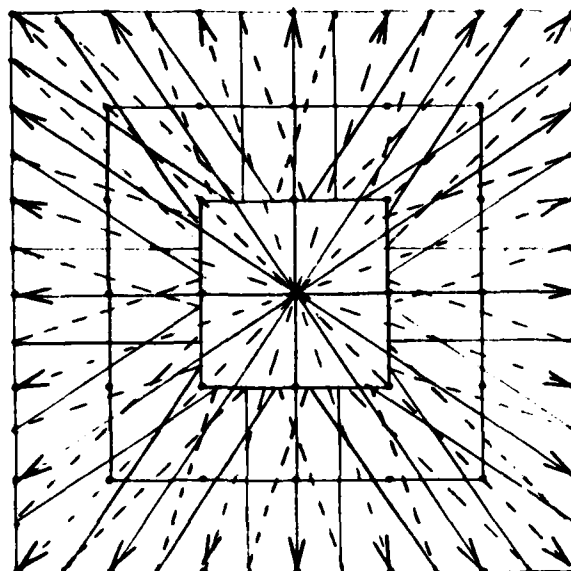
Figure II-6 Parallel Quantization Scheme Templates
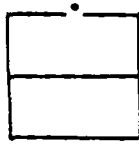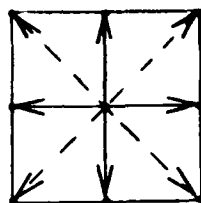(a) Ring 1, (b) Ring 2, (c) Ring 3

**Figure II-7 Generalized Form of a Link Gate Set
Parallel Quantizing Scheme**

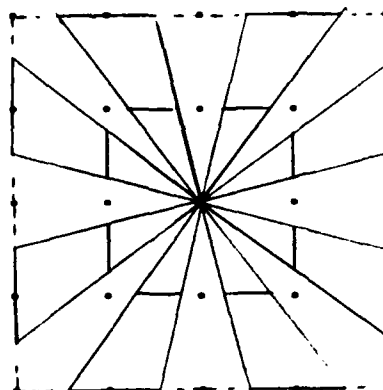## Triangular Quantization Scheme

The LGS defining the triangular quantization scheme are
formed by selecting the highest-order-ring of the code and
finding the midpoints of all adjacent nodes on the ring.
Lines are then drawn from the midpoints to the center of the
ring. These lines are referred to as "midpoint lines". The
parallel segments cut out of each ring by a pair of midpoint
lines form the LGS for the highest-order-ring. This process
is repeated for each ring contained in the code. This process
forms the template for the triangualar quantization scheme
(Ref 12:19).

The quantization procedure consists of a search for the
highest-order-ring for which all link gates intersect the
curve. The search begins by placing the center of the template
over the last node to be encoded. The LGS for the highest-
order-ring in the code is checked to determine whether all
link gates intersect the curve if not the next lower-order-
ring is checked, this process continues until a ring is found
in which all link gates intersect the curve or the lowest
order ring in the code is reached. If all link gates

intersect the curve then the node associated with that LGS is quantized and encoded and the process is repeated for the next node to be quantized and encoded. If the lowest-order-ring is reached before all link gates intersect the curve then there are two methods for selecting the node to be quantized and encoded. The first method is to find the first intersection between the line drawing and the grid lines and locate the node nearest to that intersection and quantize and encode it as the next node (Ref 10:8). The second method is to find the first intersection of the line drawing and the lowest-order-ring and select the node nearest to that intersection and quantize and encode it as the next node (Ref 10:8). Figure II-8 shows the templates associated with the triangular quantization scheme for Rings 1,2, and 3. Figure II-9 shows the generalized form of the LGS for triangualr quantization scheme.
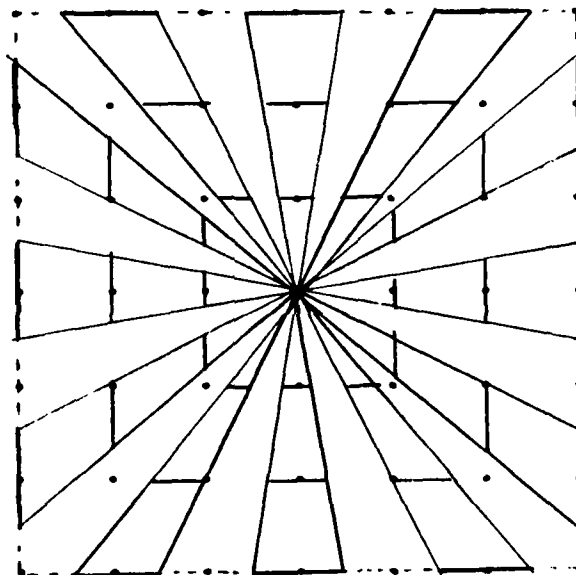
Figure II-8 Tringular Quantization Scheme Templates
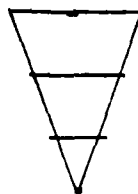(a) Ring 1, (b) Ring 2, (c) Ring 3

Figure II-9 Genral LGS for Triangular Qantization
Scheme

As can be seen from Figure II-6a and Figure II-8a the
parallel and triangular quantization schemes are the same for
single ring codes. The encoding procedure for the generalized
chain codes is essentially the same as for the 1-code. A
binary number is assigned to each node on every ring of the
code. If it is known in advance which ring will be used most
freqently then the assignment of binary numbers to nodes can
begin with that ring thereby minimizing the number of bits
required to encode a particular line drawing. In all cases
the assignment of binary numbers to the nodes must be the same
for encoding and decoding.

EVALUATION OF GENERALIZED CHAIN CODES

The standards for evaluation of the quantization of line-
drawings are very subjective. Presently there are six
suggested major criteria for evaluating the effectiveness of a
coding scheme for line-drawing data; these criteria are the
following: 1. Compactness; 2. Precision; 3. Smoothness;
4. ease of selective access; 5. ease of processing; 6.
ease of encoding and decoding (Ref 13:315). These criteria

II-14

are given different weights during evaluation depending upon the application of the quantized line drawing data.

Compactness is the amount of storage space required to store a given line-drawing or if the data is being transmitted the amount of time-bandwidth required to transmit the data. The more compact the quantization means the less storage space and transmission time-bandwidth required. However to achieve high compactness, tradeoffs must be made with the other five criteria. The more compact the quantization scheme is will lead to increased complexity in encoding, decoding, and processing (Ref 13:315).

Precision is extremely important if the line drawing data is to be used for quantitative analysis. This is extremely important if the line-drawing data is a geographic map (Ref 6:1). The precision of the quantization is highly dependent upon grid size. Thus by using a very small grid size in relation to the raduis of curvature of the line-drawing and low order codes precision can be made very high but at a cost to the other criteria, specifically compactness.

Ease of selective access refers to the relative speed with which any particular portion of a quantized line-drawing may be accessed (Ref 13:316). This criterion takes on added significance when the quantized line-drawing data represents a geographical map. Generally the access criterion leads to a decrease in compactness and hence increased storage and transmission time-bandwidth requirements (Ref 13:316).

Ease of processing refers to the simplicity and speed of

the algorithms used to quantize the line-drawing. This criterion is very dependent upon the application of the quantized line-drawing. If the line-drawing is to be stored then the ease of processing could become less important than the compactness criterion, however if the data requires frequent processing the ease of processing criterion will take precedence.

Ease of encoding and decoding become extremely important if large amounts of data are involved. For small quanities of data which requires extesisve processing the ease of processing criterion will take precedence over ease of encoding and decoding (Ref 6:1).

Quanitative Evaluation of Generalized Chain Codes

As stated previously the common means of evaluating the performance of the generalized chain codes has been very subjective. This thesis effort will utilize two of the previously stated performance criteria to quantitatively analyze the performance of the generalized chain codes. The two criteria that will be used are the precision and compactness criterion or in terms of information theory accuracy and code rate.

Accuracy in general terms refers to how closely the digitized line drawing approximates the original line drawing. To measure the accuracy of the digitized representation of the line drawing the concept of distortion is used. The distortion measure which will be used is defined as a length

normalized measure of area between the original line drawing and its quantized approximation(Ref 19:979).

The length normalized area error is obtained by summing all of the incremental areas lying between the orginal line drawing and its digitized representation and then dividing that total area by the length of the orginal line drawing in terms of grid size. Figure II-10 shows a typical line drawing that could be encoded using the 1-code and the resulting incremental area errors.
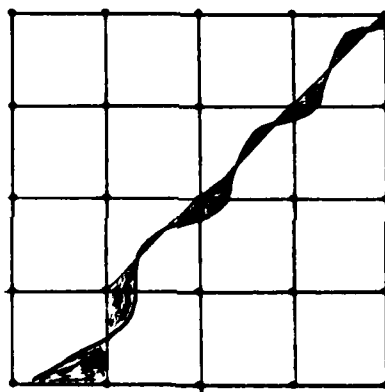

Figure II-10 The incremental area error is shown by the
shaded area

Figure II-11 shows the area error of a straight line defined by Y = X/2 which is quantized using the 1-code with grid of size q.
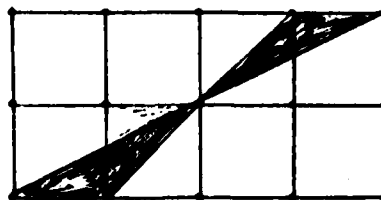
Figure II-11 Area Error of a Straight line Y=X/2
Quantized using the 1-Code. Shaded
areas are the area errors.

The area error for the straight line quantized in Figure II-11 is calculated as follows.

$$\text{Area of a Triangle} = \pm 1/2 \begin{vmatrix} X1 & Y2 & 1 \\ X2 & Y2 & 1 \\ X3 & Y3 & 1 \end{vmatrix}$$

$$= + (X1{*}Y2 + X3{*}Y1 + X2{*}Y3 - X3{*}Y2 - X2{*}Y1 - X1{*}Y3)$$

Where the sign is chosen so that the area is non negative.

$$(X1,Y1) = 0,0 \quad (X2,Y2) = 1,\tfrac{1}{2} \quad (X3,Y3) = 2,1$$

$$= + 1/2(0+0+1-2-0-0) = .5q^2$$

By symmetry it can be shown that the area of the other remaining triangles are the same. Thus for the quantization of the line drawing shown in Figure II-11 the total area error is q^2.

To find the normalized area error which is the distortion measure being used the total area error must be normalized by the length of the quantized line. For the line drawing shown in Figure II-11 the length of the line is defined by Length = $nq/\cos(\theta)$. Thus for the line drawing of Figure II-11 when quantized by a 1-code using a grid of size q the

area error per length q is .2236068. The area error per grid size for other types of line drawings can be calculated in the same manner although it is not as straight forward as it is for a straight line. A detailed discussion of the distortion measure being used can be found in references 1 and 19.

The code rate performance measure refers to the number of bits required to quantize a length q of the line drawing. This measure is obtained by determing the number of bits required to represent all of the next possible nodes. For example if a 1-code were being used 3 bits would be required and if a 1,2-code were being used 5 bits would be required to reperesent the next possible nodes. Using the 1-code to quantize the line drawing shown in Figure II-11 the code rate performance measure is defined as R = 3Cos(θ)/Nq. For the line drawing and 1-code shown in Figure II-11, R = 2.6832816 bits per length q of the original line drawing. A more detailed presentation of the code rate performance measure can be found in references 1 and 19.

SUMMARY

The concept of line-drawings and methods of processing line-drawings were introduced and discussed. The concept of Gird-Intersect Code and Generalized Chain Codes were introduced and discussed. The quantization and encoding procedures for the codes were discussed and a set of criteria for evaluating the code was presented.

Chapter III

SOFTWARE OVERVIEW

This chapter presents a brief overview of the software algorithm designed and implemented by Jones, ref 10. The algorithm as developed by Jones was designed specifically for the quantization of sine and circular waves. The quantization method implemented by Jones was the triangular scheme as described in Chapter II. The overview of the algorithm will consist of a brief description of the input parameters, the simulation of the line drawing, the calculation of the intersect points, the calculation of the nodes of quantization, and the calculation of the performance measures.

Input Parameters

The input parameters required for execution of the algorithm are: 1). The type of function to be quantized(1-sine wave, 2-circular wave); 2). The period of the function to which π x .001 is added to prevent the instaneous phase being the same in any two grid squares; 3). The amplitude of the functions if it is a sine wave(The amplitude of the circular wave is calculated by the algorithm to be 1/4 of the period); 4). The grid size is a variable input but since the amplitude and period of the functions are measured in terms of the grid size it is essentially a constant; 5). The phase shift of the functions in radians; 6). The number of periods of the function to be quantized; 7). The number of rings contained

in the quantization code; 8). The specific codes to be used in the quantization. Input 7 specifies the number of rings in the quantizing code for example the (1,3) code has two rings and input 7 would be 2. Input 8 is the rings of the code such as for the (1,3) code input 8 would consist of two entries 1 and 3.

Line Drawing Simulation

To simulate a line drawing specified by a function, the following must be defined: The function; its inverse; and its derivative.These functions were implemented by Jones in his algorithm using the functions F, DF, FINV respectively. The detailed descriptions of these functions can be found in ref 10 pages 22 - 25.

Calculation of Grid Intersects

The calculation of the grid intersects is accomplished through the use of the two subroutines FIP and CIBD.

FIP When the grid intersects are to be calculated the subroutine FIP is called by the main program. The FIP subroutine requires three inputs for execution. These three inputs are the location of the first grid intersect to be calculated, the subscript of the first available storage location in the grid intersect array, and the largest subscript of the grid intersect array. The largest subscript of the grid intersect array is calculated in the main program as one less than the dimensions of the array minus twice the amplitude of the function being quantized(Ref 10:26). The

other inputs to the subroutine are zero and one for the first time the subroutine is called by the main program. The zero indicates the position on the X axis and the one is the first subscript of the grid intersect array. The subroutine uses a flag to indicate to the calling program whether it has calculated all of the grid intersects of the line drawing prior to returning. The FIP subroutine only calculates the vertical grid line intersects. Subroutine CIBD is called to perform the calculation of the horizontal grid line intersects. In addition to calling CIBD subroutine FIP calls functions F, DF to calculate the vertical grid intersects (Ref 10:26-30).

CIBD The CIBD subroutine is used to calculate the horizontal grid intersects within a certain interval. The inputs to CIBD are the ordinate values of the end points of the interval, and the derivative of the function at the end points of the interval. CIBD calls function FINV to determine the value of the abcissa for a particular ordinate value(Ref 10:30-31) .

Node of Quantization Calculation

These calculations are performed by the set of five subroutines: GERD; FNSRI; FNGN; FAR; and ANGLE. These subroutines are independent of the line being quantized. These subroutines are dependent upon only the quantization scheme being used to quantize the line drawing, which in this case is the triangular scheme which was described in Chapter II. The subroutine GERD performs the actual quantization of

the line drawing utilizing FNSRI, FNGN, FAR, and ANGLE to perform associated quantizing tasks.

GERD   The input to GERD is the flag set by subroutine FIP indicating whether or not all intersect points for the line drawing have been calculated. GERD reads the intersect points calculated by FIP from the intersect point storage array and calculates the encoded nodes. The encoded nodes are stored in an array with the same dimensions as the intersect point array. GERD calls subroutines FNGN, FNSRI, FAR, and ANGLE to perform the calculation of the encoded nodes(Ref 10:32-34).

FNSRI   This subroutine is used to find the first grid intersect point lying on the code ring. The inputs to this subroutine are the level of the code ring being considered, and the subscript of the last encoded node which is the center of the ring. The subroutine returns either the subscript of the grid intersect that lies on the ring or a zero indicating that it did not find a grid intersect point lying on the ring(Ref   10:34).

FNGN   This subroutine calculates the coordinates of the grid node lying closest to a grid intersect point. The input to to this subroutine is the subscript of the grid intersect point for which the node coordinates are to be calculated. This subroutine returns the values of the X and Y coordinates of the grid node lying closest to the inputted intersect point(Ref 10:35).

FAR   This subroutine calculates the midpoints of the grid

nodes. The inputs to this subroutine are the subscript of the node lying in the center of the ring, the code level being considered, and the coordinates of the grid node for which the midpoints are to be calculated. The subroutine returns the angles associated with the midpoints and the center of the ring and a flag indicating in which quadrant the angles are located. This subroutine calls subroutine ANGLE to calculate the angle subtended by a line drawn from the center of the ring to a midpoint and the horizontal grid line of the center of the ring(Ref 10:35-36).

ANGLE This subroutine calculates the angle subtended by a line drawn from the center of a ring to a point on the ring and the horizontal grid line of the center of the ring. The inputs to this subroutine are the subscript of the center of the ring and the coordinates of a point on the ring. The subroutine returns the angle subtended in radians(Ref 10:37).

Performance Measure Calculations

The calculation of the performance measures requires the use of eight subroutines. These subroutines are: CAE; CAESL; FDE; DFDE; ZERO; CLI; FDLI; and SPGQI. These subroutines are used to calculate the area error and path length of the line drawing for some arbitrary interval.

CAE This subroutine divides the line drawing into intervals corresponding to the line segments which are used to approximate the line drawing and is the first subroutine called when error calculations are to be performed. The

III-5

inputs to this subroutine are the end points of the interval for which the area error is to be calculated. The subroutine returns the area error for the interval. The subroutine calls subroutine CAESL(Ref 10:38-39).

CAESL This subroutine divides the intervals given to it by subroutine CAE into intervals for which the integrand defining the area error is strictly positive or negative. The inputs to the subroutine are the end points of the intervals calculated by subroutine CAE. The subroutine returns the area error for the interval inputted. The subroutine calls subroutines ZERO and SPGQI(Ref 10:39).

ZERO This subroutine is used to find the zero of an inputted function within a given interval. The subroutine uses the modified regula falsi algorithm to calculate the zero of the given function. A detailed description of the modified regula falsi algorithm can be found in reference 21 chapter 2. The inputs to the subroutine are the function for which the zero is to be found and the end points of the interval in which a zero lies. The subroutine returns the value of the X coordinate for which the inputted function has a zero. The inputted functions are either FDE or DFDE(Ref 10:39).

SPGQI This subroutine uses the six-point Gaussian Quadrature Iteration Algorithm to perform the numerical integration of a particular function over a given interval. A complete discussion of numerical integration can be found in references 21 - 26. The inputs to the subroutine are the function to be

integrated and the end points of the interval of integration.

The subroutine returns the value of the integral(Ref 10:40).

FDE   This function evaluates the integrand defining the area

error for some given time.   The input to the function is the

particular value for which the integrand is to be evaluated.

This   function   calls   function DF which calculates   the

derivative of the function defining the line drawing at   the

inputted time.   This function is an inputted function to

subroutines ZERO and SPGQI.

DFDE This function evaluates the derivative of the integrand

defining the area error for some given time. The input to the

function is the particular value for which the derivative is

desired.   This   function calls function DF which calculates

the   derviative of the function defining the line drawing at

the   inputted time.   This function is an inputted function to

subroutines ZERO and SPGQI.

CLI   This subroutine calculates the path lenth of the line

drawing over an interval.   The inputs to the subroutine are

the end points of the interval.   The output of the subroutine

is   the   path length of the particular line drawing@n the

inputted interval.   This subroutine calls function FDLI

(Ref 10:40).

FDLI   This function calculates the value of the integrand

definding the path length of the line drawing.   The input to

the function is the particular value for which the integrand

is to be evaluated.   This function is an inputted function to

III-7

SPGQI. This function calls function DF which calculates the derivative of the function defining the line drawing for the inputted value.

Summary

This chapter provided a brief overview of the algorithm developed and implemented by Jones and used in this thesis.

Chapter IV

Algorithm Modifications

The algorithm developed and implemented by Jones reference 10 required modification to allow the selection of a grid intersect lying within one-half of a grid size of the ring to be considered on the ring for quantization. The Jones algorithm was designed to select only those intersect points which were lying exactly on the ring for quantization. The algorithm also required extensive modification to implement the quantization of a rotated sine wave. The modification required to implement the selection of a grid intersect lying within one-half of a grid size will be discussed first since it was a slight modification.

Grid Intersect Selection Modification

This modification only required the alteration of the subroutine FNSRI. The subroutine as written by Jones required a grid intersect to lie exactly on the ring to be considered as being on the ring for quantization purposes. This subroutine was modified to allow a grid intersect lying within one-half of a grid size to be considered to be on the ring. The area in which a grid intersect must lie and still be considered as on the ring for quantization is called the capture region. Thus, as implemented by Jones, the capture region was zero. The modification yields a capture region of one-half grid size and the region is a ring of one grid size in depth centered on the encoding ring. Figure IV-1 shows an example of the capture region for a (2) code ring.
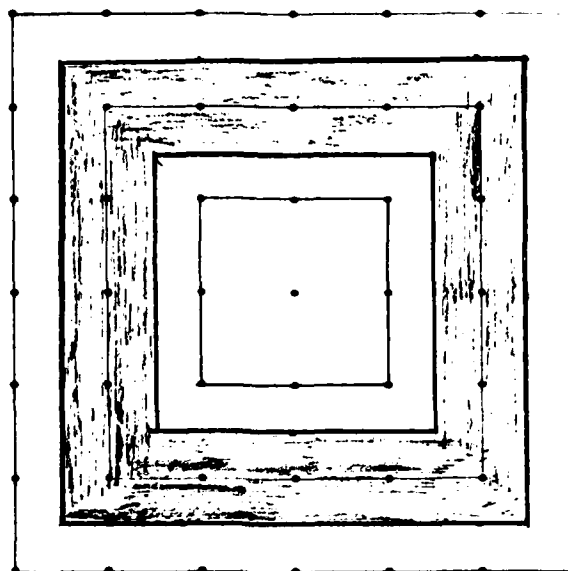
Figure IV-1   Capture Region for Modified Algorithm

The   orginal lines of code in subroutine *FNSRI* are as shown in Figure   IV-2A and the modified lines of code are as   shown   in Figure IV-2B.

```
IF(XDIS.EQ.DVR) KXT=0
IF(YDIS.EQ.DVR) KYT=0
```

Figure IV-2A   Orginal Code


```
IF((XDIS.GT.DVR-.5*DEL).AND.
IF(XDIS.LT.DVR+.5*DEL)) KXT=0
IF((YDIS.GT.DVR-.5*DEL).AND.
IF(YDIS.LT.DVR.+.5*DEL)) KYT=0
```

Figure IV-2B Modified Code

Rotated Sine Wave Modification

The implementation of the rotated sine wave required extensive modification of the Jones algorithm. The initial effort to rotate the sine wave was by defining the rotated sine wave by the function $Y = \text{Tan(angle of rotation)} + \text{Amplitude} * \text{Sine}(2*\pi*T+\Theta)$. Further analysis revealed that this function was not a true rotation of the sine wave but was rather a translation of the sine wave. After a search of the literature it was determined that a point by point rotation would have to be performed to rotate the sine wave. Also it was realized that the rotation produced a multivalued function and hence the new function would not be uniquely invertible. Further complicating the problem was that an exact mathematical expression for the function mapping the rotated points could not be determined. The functions performing the rotation of the points are as shown in Figure IV-3(Ref 20:80).

$$X = x * \text{Cosine(rota)} - y * \text{Sine(rota)}$$
$$Y = x * \text{Sine(rota)} + y * \text{Cosine(rota)}$$

rota = angle of rotation X,Y are new coordinates
defined in relation to original axis

Figure IV-3 Point Rotation Functions

Based upon the problems outlined above the assumptions made in Chapter I have to be modified since the line drawing cannot be described by an exact mathematical function. The

IV-3

problem of not being able to describe the function that maps the rotated points also means that the inverse of the function defining the line drawing and the derivative of the function defining the line drawing cannot be described by an exact mathematical expression. These problems required the following modifications be made on Jones algorithm to quantize a rotated sine wave. The discussion of the modifications will follow the order in which they are encountered in the Jones algorithm.

Main Program The first modification required was to add the angle of rotation to the inputs and outputs of the algorithm. The second modification required is the calculation of the path length of the wave form over one period. Since the sine wave is rotated at some angle of rotation above the X-axis the period of the sine wave without rotation will be related to the period of the rotated sine wave by the cosine of the angle of rotation. The new starting point of the sine wave must also be determined. Using the initial value of zero for the starting point of the sine wave on the X-axis the rotated sine wave will have a starting point of the negative amplitude of the orginal sine wave multiplied by the sine of the angle of rotation. The orginal lines of code and the modified lines of code in the main program are as shown in Figures IV-4A and B respectively.

```
      IF(NF.EQ.1) READ(*,*) A,PER,THETA
            WRITE(*,120) A,PER,THETA
120 FORMAT (25X,5H A: , E17.10/25X,7HPER: ,E17.10/
    + 25X,9H THETA: ,E17.10)

            CALL CLI(0,PER,VLIPP)
```

Figure IV-4A   Original Code

```
      IF(NF.EQ.1) READ(*,*) A,PER,THETA,ROTA
            WRITE(*,120) A,PER,THETA,ROTA
120 FORMAT (25X,5H A: ,E17.10/25X,7HPER: , E17.10/
    + 25X,9H THETA:,E17.10/25X,10HROTATE: ,E17.10)


            G = F(0.0)
        P = - G * SINE(ROTA)
      PER = PER * COSINE(ROTA)
            TS = 0.0
            LNIP = 1
          LENIP = 100
      CALL FIP(TS,LNIP,LENIP,JFIN)
            LB = 1
            ST = 0.0
      CALL CLI(ST,PER,VLIPP)
```

Figure IV-4B Modified Code

Find Intersect Points

The problem of finding the intersect points for a line
drawing which is generated by the point to point rotation of
another line drawing required the writing of four new
subroutines. The Jones algorithm used the knowledge of the
inverse of the function defining the line drawing to calculate
the horizontal gird intersect points. Since the function
mapping the rotated points could not be described in an exact
mathematical sense neither could its inverse. To overcome

this problem a line following procedure was used to calculate the intersect points. This technique requires repeated evaluations of the function F which generates the sine wave on the X-axis and the subroutine ROTATE which performs the point by point rotation of the values inputted to function F and returned by function F. The line following technique also require continous checking to determine if a gridline has been crossed and in what direction the crossing was made. These tasks were implemented by the subroutines FIP, ROTATE, and GCROSS and by function F. Function F did not require any modification and will not be discussed.

FIP The inputs and outputs of this subroutine are the same as described in Chapter II. The first step required by the subroutine is to establish accounting variables which are the same as described by Jones, reference 10 pages 26 - 30. The next step performed by FIP is to determine if the variable TSTART is equal to zero. If TSTART is equal to zero the subroutine INITPT is called. Subroutine INITPT ensures that that the first grid intersect to be encoded lies in the right half plane of the cartesian coordinate system. If INITPT is called it will calculate the first grid intersect point and increment the pointer to the grid intersect array. If subroutine INITPT is not called then the subroutine ROTATE will be called. The subroutine calculates the coordinates of the rotated points for a particular value inputted. The variables RT and RFT are the new X and Y coordinates

calculated by subroutine ROTATE and are returned to subroutine
FIP by the common block CR. The variables RT and RFT are then
stored as the first intersect points and the intersect array
pointer NIP is incremented. The next step is to initialize
the variables IX and IY. These variables are set equal to the
integer values of RT and RFT respectively. These variables
are used in the calculation of grid line crossings. The value
T1 is then incremented by two tenths of a grid size and
renamed as T2. The subroutine ROTATE is called to calculate
the new values of RT and RFT. After returning from subroutine
ROTATE the subroutine GCROSS is called to determine if a grid
line has been crossed. If a grid line has not been crossed
the variable T2 is incremented by two tenths of a grid size
and the process is repeated until a grid line crossing is
detected by subroutine GCROSS. If a grid line crossing is
detected by GCROSS it returns a flag value greater than zero
to subroutine FIP. The variable T is then set equal to the
variable T2 minus two tenths of a grid size. A binary search
algorithm is then initiated to determine the value of T for
which the grid line crossing occurred. The value of T chosen
is within .00625 of the actual value for which the grid
crossing occurred. Subroutines ROTATE and GCROSS are called in
the binary search. The next step is to store the values of RT
and RFT calculated for the last value of T in the intersect
array and increment NIP. The next step is to check the flag
value returned by subroutine GCROSS and determine the new
values of variables IX and IY. The new values are determined

IV-7

by the value of the flag passed from subroutine GCROSS via the common block CR. The values of both IX and IY can be changed in either negatively or positively, simultaneously or separately depending upon the value of the flag set by GCROSS. This process continues until the end of the line drawing is reached or until the varaible NLIMIT is reached. NLIMIT is calculated in the main program as described in Chapter III. If NIP exceeds NLIMIT prior to calculating all of the grid intersect points of the line drawing the variable JFIN is set equal to zero indicating to the main program that there are more intersect points to be calculated. TSTART is set equal to the last value of T2 and NIP is set equal to NIP -1 and a return is made to the main program.

ROTATE   This subroutine is called by subroutine FIP and calls function F. This subroutine uses the expressions of Figure IV-3 to calculate the rotated points for an inputted value of X. The subroutine calls function F to calculate the value of the line drawing on the X-axis and then performs the transformation of the points to generate the rotated sine wave. The coordinates of the rotated points are then passed to subroutine FIP via the variables RT, and RFT through the common block CR.

INITPT   This subroutine is called by subroutine FIP to ensure that the first grid intersect to be calculted lies in the right hand plane of the cartesian coordinate system. INITPT calls the subroutine ROTATE to evaluate the value of variable

RT to determine if RT is greater than or equal to zero. If RT is equal to zero the values of RT and RFT are stored as the coordinates of the first intersect point and NIP the pointer to the intersect point storage array is incremented. The variables IX and IY are then initialized. IX is initialized as zero and IY is initialized as the integer value of variable RFT. T1 is then set equal to TSTART and a return to FIP is made. If RT is less than zero TSTART is incremented by one grid size and subroutine ROTATE is called to calculate the associated value of RT, this process continues until RT is greater than or equal to zero. If a value of TSTART is found for which RT is equal to zero the initialization process described above is followed and a return is made to FIP. If RT is greater than zero a binary search algorithm is initiated to find the value of TSTART which will ensure that the first intersect point to be stored lies in the right hand plane. Once the value of TSTART has been determined the variables IX and IY are initialilized as described above. The values of RT and RFT are stored in the intersect point array and the intersect point array pointer NIP is incremented. T1 is set equal to TSTART and a return to FIP is made.

GCROSS   This subroutine is called by subroutines FIP and INITPT to determine whether a grid line crossing has occurred. The inputs and outputs of the subroutine are through the common block CR. The first action the subroutine takes is to initialize the flags indicating whether a grid line has been crossed. These flags are IFLAGX and IFLAGY which indicate

whether the vertical or horizontal grid line has been crossed respectively. The next step is to determine which grid lines can be considered as possible candidates to be crossed. This is accomplished by determining whether IX and IY are positive or negative and then either adding or subtracting DEL to IX and IY to initialize the variables IXU,IXL,IYU,IYL. The variables determine the upper and lower bounds for grid line crossings for the vertical and horizontal gridlines respectively. The next step is to determine if a grid line has been crossed and which grid line was crossed. If RT is greater than IXU or less then IXL a vertical grid line has been crossed. IF RFT is greater than IYU or less than IYL a horizontal grid line has been crossed. If a grid line has been crossed the flags, IFLAGX and IFLAGY, are set to indicate which grid line was crossed. The next step is to set IFLAG equal to the sum of IFLAGX and IFLAGY. If the sum is greater than zero a grid line crossing has occurred. The value of IFLAG is passed to FIP via the common block CR. A return to FIP is then made.

Node of Quantization Calculation

This section of the Jones algorithm required only one modification to perform the quantization of the rotated sine wave. The subroutine FNSRI had to be modified to allow grid intersects that did not lie exactly upon the ring to be considered on the ring. This modification was performed in the same manner as described in Chapter III. The capture

IV-10

region for the ring was set at two tenths of the grid size rather than half the grid size as described in Chapter III.

Performance Parameter Calculations

An extensive amount of modification was required in the subroutines that calculate the performance parameters. These modifications were required due to the fact that an exact mathematical expression could not be developed that would map the points of the rotated sine wave. Since an exact expression for the points defining the rotated sine wave could not be developed it was necessary to generate a function that would approximate a function that would map the points of the rotated sine wave. The modifications made to the subroutines and functions of the Jones algorithm will be discussed first and then the subroutine used to generate the approximation of the function defining the rotated sine wave will be discussed.

FDE The modification made to function FDE was slight. The modification was required due to the inability to describe the line drawing being quantized by an exact mathematical expression. The original code and the modified code are as shown in Figures IV-5 A and B respectively.

$$FDE = F(X) - S*(X-XGN(LB)) - YGN(LB)$$

Figure IV-5A Original Code Function FDE

$$FDE = APROX(X) - S*(X-XGN(LB)) - YGN(LB)$$

Figure   IV-5B Modified Code Function FDE

<u>DFDE</u>   Since the points defining the rotated sine wave could
not be described by an exact mathematical expression the
derivative also can not be found directly.   To find the
derivative at a certain point numerical differentiation was
used.   To   minimize   the   error   inherent   in   numerical
differentiation the following expression was used to evaluate
the derivative at a given point. $f(x) = (F(x + h) - F(x - h))/(2. * h)$ (Ref 21:278).   The value of h must be chosen
sufficiently small to provide reasonable accuracy in the
evaluation of the expression for some value of X.   In this
function the value of h was chosen to be .003.   Figures IV-6A
and B show the orginal code and the modified code
respectively.

$$DFDE = DF(X) - S$$

Figure IV-6A Original Code Function DFDE

```
H = 3.E-3
Z = X + H
W = X - H
DFDE = (((APROX(Z) - APROX(W))/(2.*H)) - S
```

Figure IV-6B Modified Code Function DFDE

<u>FDLI</u>   The modification made to this function is similiar to

that made in function DFDE. Numerical differentation is also required since the derivative of the function defining the line drawing is required. The orginal code and the modified code is as shown in Figures IV-7A and B respectively.

$$FDLI = SQRT(1. + DF(X)*DF(X))$$

Figure IV-7A Original Code Function FDLI

$$H = 3.E-3$$
$$Z = X + H$$
$$W = X - H$$
$$DF = (APROX(Z) - APROX(W))/(2. * H)$$
$$FDLI = SQRT(1. + DF*DF)$$

Figure IV-7B Modified Code Function FDLI

APROX   Function APROX is based upon the Weierstrass Theorem(Ref 25:26). The Weierstrass Theorem states that for some continous function on a closed bounded interval that there exists some polynomial, p, such that for all X in the closed bounded interval the inequality $f(x) - p(x) \leq \varepsilon$ holds(Ref 25:26). Using the results of th Weierstrass Theorem it has been shown that there exist a nique polynomial which satisfies the Weierstrass Theorem(Ref ₋ J: 29-30). There are several forms of the polynomial which can be used to interpolate the fuction. Some of the more common forms are the La Grange(Ordinate form), Newton's Interpolating Polynomials(Difference form), and the Aitken-Neville Repeated (Iterated) Linear Interpolation (Ref 24:203-238).

IV-13

The La Grange interpolating polynomial was seleted to provide the aproximation of the line drawing as it could be used when the abcissa values are arbitary but distinct(Ref 24:233) A detailed discussion of the La Grange interpolating polynomial algorithm can be found in reference 24. A discussion of how the algorithm was implemented is provided.

La Grange Interplolating Polynomial Implementation - The input to the function APROX is the value of X for which the approximation is desired. The first step of the function is to initialize control variables. The first variable intialized is NP1 which is set as five which means that five points will be used by the approximating algorithm and the order of the approximating polynomial will be four. The next variable initialized is RAN which is set equal to the inputted value of X minus one. This value is chosen to ensure that the inputted value will be within the interval of the points used in the approximation. The variable L is set equal to the variable LB in common block CAEP. This variable is used as a pointer to the intersect points stored in the intersect point array. The next step is to determine if RAN is less than zero. If RAN is less than zero the first five intersect points will be used by the approximating algorithm. If RAN is greater than zero a comparison of the X coordinates of the intersect points stored in the intersect point array pointed to by L is performed until an intersect point is found which is greater than or equal to RAN. A test is then performed to

ensure that the value of NIP will not be exceeded during the
execution of the La Grange interpolating polynomial algorithm.

Summary

In this chapter the modifications to the Jones algorithm
required to allow the selection of grid interect lying within
one-half of a grid size of the ring and the quantization of a
rotated sine wave were discussed.

Chapter V

Performance Analysis

In this chapter the performance of the generalized chain codes will be analyzed. The performance of the individual codes will be examined and then their performances will be compared. Last the effect of enlarging the capture region will be analyzed and then the performance of the algorithm used to quantize the rotated sinusoids will analyzed.

The performance of the codes will be evaluated using the parameters of area error and code rate as described in chapter II. Appendix A contains all of the figures and tables referenced in this chapter pertaining to performance of the codes.

Individual Code Performance

The (1), (1,2), (1,3), and (1,2,3) codes were selected for analysis and were used to quantize identical sinusoids. Four sets of sinusiods were quantized to generate the data for the analysis. Set one was sinusoids with a constant period of 10 and amplitudes varying from 2 to 100. Set two was sinusiods with a constant period of 20 and amplitudes varying from 5.1 to 6.0. Set three was sinusoids of period 20 and amplitudes varying from 5 to 60. Set four was a set of sinusoids of periods 10 and 20 with amplitudes of 500 and 1000 for each period. The performance of the individual codes will be analyzed with respect to their ability to quantize the four sets of sinusoids as described above.

(1) Code This code experienced a wide flucuation in the area error and code rate performance. Figures A-1 thru A-4 show the performance of the (1) code quantizing the sinusoids of period 10 and amplitudes varying from 2 to 100. The area error performance was generally poor for this code as expected due to the limited angular resulotion of the code. The best area error performance was for amplitude of 2 and then as the amplitude was increased the area error increased. The best code rate was also for amplitude 2 and again as the amplitude increased the code rate increased. It was noted that the area error appears upperbounded by .25 and the code rate by 3.0.

The quantization of the sinusoids with a period of 20 and amplitudes varying from 5.1 to 6.0 produced some surprising results. Figures A-5 and A-6 portray the performance of the (1) code quantizing the sinudoids. The data generated by the quantization of the sinusoids of period 20 and amplitudes of 5.1 to 6.0 indicates that the position of the point of zero slope of the sinusoid with respect to the horizontal grid lines has a direct influence on the area error performance of the code. This was checked to determine if this performance was just a quirk but the quantization of sinusoids of period 20 and amplitudes varying from 10.0 to 11.0 and 20.0 to 21.0 produced similiar results as shown by Table A-2. The code rate did not change appreciably as was expected for such a small change in amplitude.

Figure A-7 depicts the performance of the (1) code when quantizing sinusoids of period 20 and amplitudes ranging from

V-2

5.0 to 60.0. The performance of the code again decreased as the amplitude increased. Again it was noted that the code rate was upper bounded by 3.0 and the area error performance was upper bounded by .25.

The quantization of the sinusoids of periods 10 and 20 with amplitudes of 500 and 1000 revealed some interesting results. The asymptotic code rate and area error for these sinusoids are depicted in Table A-1 and again it appears that the area error is uppper bounded by .25 and the code rate is upper bounded by 3.0. Since the amplitudes of these sinusoids were extremely high as compared to the periods, it can be assumed with little loss of accuracy that the waveform looks like a straight line within any single grid square. If this approximation is made, then it can be shown graphically that this is the highest code rate and area error that can be expected from the (1) code.

The (1) code as discussed above degrades very rapidly as the amplitude to period ratio is increased. There also seems to be a very strong link between the position of the point of zero slope of the sinusoid with respect to the horizontal grid lines of the qunatizing grid and the area error performance of the (1) code as illustrated by Figures A-5 thur A-6 and Table A-2.

(1,2) Code The analysis of the data generated by the (1,2) code quantizing the sets of sinusoids produced some surprising results. Figures A-8 thru A-11 depict the

performance of the (1,2) code quantizing sinusoids with periods of 10 and amplitudes varying from 2 to 100. The area error of the (1,2) code increased as the amplitude increased. The code rate of the (1,2) code increased dramatically when the amplitude was increased from 2 to 4. The code rate decreased to about the rate for amplitude 2 for amplitudes 6,8, and 10 as shown in Figure A-8b. The code rate exhibited a large increase as the amplitude was increased from 10 to 14 and then changed minimally as the amplitude was increased from 14 to 16 as shown by Figure A-9a. The code rate exhibited a steady increase as the amplitude was increased from 16 to 100. These results were surprising since it was anticipated that the code rate and area error would decrease as the amplitude to period ratio increased. This was anticipated since as the amplitude to period ratio increased the waveform being quantized would begin to look more and more like a straight line within any two adjacent grid squares. Thus it was anticipated that the nodes lying of the second ring would be selected with higher frequency as the amplitude to period ration increased. The area error for this set of sinusoids seemed to be upper bounded by .24 and the code rate upper bounded by 3.60 as depicted by Figure A-11.

The performance of the (1,2) code when quantizing sinusoids of period 10 and amplitudes varying from 5.1 to 6.0 is as shown by Figures A-12 and A-13. The performance of the code degraded as the position of the point of zero slope of the waveform was moved away from a horizontal grid line. The

V-4

code rate and area error exhibited large changes for small changes in amplitude as the amplitude was increased from 5.1 to 5.5 as shown by Figure A-12 . The area error decreased as the amplitude was increased from 5.6 to 6.0 as shown by Figure A-13b. The code rate flucuated as the amplitude was increased from 5.6 to 6.0 with the lowest code rate occurring for an amplitude of 5.7 as shown by Figure A-13a .

The performance of the (1,2) code quantizing the sinusoids of period 20 and amplitudes varying from 5 to 60 is as shown by Figure A-14. The area error and code rate flucuated for these sinusoids. The lowest area error occurred for an amplitude of ten and then for further increases in amplitude the area error increased as shown by Figure A-14b. The lowest code rate occurred for an amplitude of 20 and then exhibited large increases as the amplitude was increased as depicted by Figure A-14a .

The (1,2) code when used to quantize sinusoids of periods 10 and 20 with amplitudes of 500 and 1000 generated some surprising data. The code rate was approximately 3.7 and the area error was approximately .25 as depicted by Table A-2. Using the assumption that for such high amplitude to period ratios that the waveform looks like a straight line in any two adjacent grid squares an approximation can be made as to the frequency that the ring 2 nodes are selected for quantization. Knowing that it takes five bits to encode a node for the (1,2) code whether the node is lying on ring one or ring two, the

maximum code rate that could be exprienced for the quantization of a straight line is 5.0 and the minimum is 2.5. Using these results it is estimated that the nodes on the second ring were selected less than 1/3 of the time. This result also indicates why the area error performance approaches that of the (1) code since the performance characteristics of the (1) code dominate the expected advantages of the (1,2) code.

The (1,2) code as discussed above degrades in performance as the amplitude to period ratio is increased. The code also seems to be dependent upon the position of zero slope of the waveform being quantized with respect to the horizontal grid lines of the quantizing grid and the area error performance.

(1,3) Code The performance of this code exhibited a general decline as the amplitude to period ratio increased. Figures A-15 thru A-18 depict the performance of the (1,3) code quantizing the sinusoids of period 10 and amplitudes ranging from 2 to 100. The code rate decreased as the amplitude was increased from 2 to 10 as shown by Figure A-15a. The area error performance flucuated with the lowest area error occurring for an amplitude of 6 and the highest area error occurring for amplitudes of 4 and 8. The area error then demonstrated a steady increase as the amplitude was increased to 100. The code rate did not exhibit any appreciable change when the amplitude was increased from 10 to 12 and then exhibited a definite increase as the amplitude was increased to 16, as depicted by Figure A-16a . The code rate

V-6

remainded constant for amplitudes of 16, 18, and 20 as shown by Figure A-16a and then exhibited a steady increase as the amplitude was increased.

The performance of the (1,3) code quantizing sinusoids of period 20 and amplitudes ranging from 5.1 to 6.0 is as shown by Figures A-19 thru A-20. The area error of the code increased as the amplitude was increased from 5.1 to 5.5 and then decreased as the amplitude was increased to 6.0, as shown by Figures A-19b and A-20b . The code rate did not change appreciably in either direction as it flucuated with the lowest rate occurring for an amplitude of 5.8 and the highest rate occurring for an amplitude of 5.1, as shown by Figures A-19a and A-20a .

The performance of the (1,3) code quantizing the sinusoids of period 20 and amplitudes varying from 5 to 60 is depicted by Figure A-21 . The area error decreased slightly when the amplitude was increased from 5 to 10 and then showed a steady increase as the amplitude was increased to 60 as shown by Figure A-21b. The code rate exhibited a decrease as the amplitude was increased from 5 to 20 and then the code rate began to increase as the amplitude increased as is shown by Figure A-21a .

The performance of the (1,3) code was not as expected when quantizing the sinusoids of periods 10 and 20 and amplitudes of 500 and 1000. The code rate was approximately 3.9 and the area error approximately .25 as shown by Table A-1. Using the

straight line approximation as used for the (1,2) code an estimate of the frequency of selection of the nodes on the third ring for encoding can be made. The minimum code rate that could be anticipated for a straight line along the X-axis is 1.66 and the maximum is 5.0. Using the assumption that the waveform at such a high amplitude to period ratio looks like a straight line in any three adjacent grid squares the nodes lying on the third ring are being selected for encoding about 11% of the time. The low frequency of selection of the nodes on the third ring does not allow the code to take advantage of finer angular resulotion obtainable through the use of the nodes on ring three. Thus since the nodes of ring one are selected with by far the greatest frequency the area error performance approaches that of the (1) code.

The (1,3) code exhibited a general decline in performance as the amplitude to period ratio increasd. The area error performance of the code seems to be very dependent upon the position of the point of zero slope of the sinusoid with respect to the horizontal grid lines of the quantizing grid. This concept was tested for sinusoids of period 20 and amplitudes of 10.0 to 11.0 and 20.0 to 21.0 and the similiar results were obtained as depicted by Table A-3.

(1,2,3)Code The performance of the (1,2,3) code quantizing the sinusoids of period 10 and amplitudes of 2 to 100 is as shown by Figures A-22 thru A-25. The code rate exhibited a general decrease as the amplitude was increased from 2 to 10 as is shown by Figure A-22a. The area error performance

flucuated with the lowest area error occurring for an amplitude of 2 and the highest for an Amplitude of 4 as shown by Figure A-22b . The code rate and area error both increased as the amplitude increased from 10 to 100 as shown by Figures A-23 thru A-25 .

The performance of the (1,2,3) code when quantizing sinusoids of period 20 and amplitudes ranging from 5.1 to 6.0 is as shown by Figures A-26 and A-27. The code rate increased significantly as the amplitude was increased from 5.2 to 5.5 as shown by Figure A-26a. The area error increased significantly as the amplitude was increased from 5.3 to 5.5 as shown by Figure A-26b. The code rate decreased significantly as the amplitude was increased from 5.5 to 5.7 and then exhibited a slight increase as the amplitude was increased from 5.7 to 6.0 as shown by Figure A-27a. The area error decreased significantly as the amplitude was increased from 5.5 to 5.6 and remained fairly constant as the amplitude was increased to 6.0 as shown by Figure A-27b .

The performance of the (1,2,3) code when quantizing the sinusoids of period 20 and amplitudes ranging from 5 to 60 is as shown by Figure A-28. The code rate for amplitudes 5 and 10 was almost the same, a decrease was exhibited when the amplitude was increased to 20 and then the code rate increased as the amplitude increased as is shown by Figure A-28a. The area error decreased slightly as the amplitude was increased from 5 to 10 and then the area error increased as the

amplitude increased as is shown by Figure A-28b .

The performance of the (1,2,3) code when quantizing the sinusoids of period 10 and 20 with amplitudes of 500 and 1000 was surprisingly poor. The code rate and area error were approximately 4.1 and .25 respectively as shown by Table A-1. Using the approximation that for such a high amplitude to period ratio the waveform would look like a straight line in any three adjacent grid squares an approximation can be made as to the frequency that the nodes on either ring one, two or three are selected can be made. The code rate for a straight line lying on the X-axis when quantized by the (1,2,3) code would be a maximum of 6 and a minimum of 2. Using these approximations the frequency of selection of ring three nodes is 11%, ring two nodes are selceted 33% of the time, and ring one nodes 56% of the time. Again it is noted that since the ring one nodes are selected with the highest frequency the area error performance of the code is dominated by the performance of the (1) code.

The (1,2,3) code exhibited a general decline in performance as the amplitude to period ratio increased. The code also exhibited a very strong dependence between the position of the point of zero slope of the wavefrom and the horizontal grid lines of the quantizing grid and the area error performance of the code.

Comparision of Code Performance

The performance of the various codes will now be compared with respect to their ability to quantize identical sinusoids.

Two sets of data were generated in addition to the data generated and discussed above for the comparision of the codes. Set five consists of a set of sinusoids of period 20 and amplitudes varying from 5 to 160. Set six is a set of sinusoids of amplitude 5 and periods varying from 5 to 50.

Figures A-29 thru A-35 depict the performances of the codes when quantizing sinusoids of period 20 and amplitudes varying from 5 to 160. The area error performances of the (1,3) and (1,2,3) codes were almost equal and in all cases they were the lowest. The (1) code was consistently the worst performer with respect to area error performance. The (1,2) code performed consistently better than the (1) code but worse than the (1,3) and (1,2,3) codes. The code rate performances of the codes flucuated. The (1) code had the best code rate for the sinusoid of amplitude of 5 and period 20 as shown by Figure A-29a. The (1,2) and (1,2,3) codes had a slightly higher code rate and the (1,3) code had a much higher code rate. For an amplitude of 10 and period of 20 the (1,3) code had the lowest code rate followed by the (1,2,3), (1), and (1,2) codes as shown by Figure A-30a. The (1,3) code again had the lowest code rate for the sinusoid of amplitude 20 and period 20, followed by the (1,2,3), (1,2), and (1) codes as shown by Figure A-31a. The (1,3) code had the lowest code rate for the sinusoid of period 20 and amplitude 40 followed by the (1,2,3), (1), and (1,2) codes as depicted by Figure A-32a. When the amplitude was increased to 60 the (1) code had

the lowest code rate followed closely by the (1,3) code as is shown by Figure A-33a. As the amplitude was increased the (1) code was consistently the best performer followed by the (1,3) code and then the (1,2) or (1,2,3) codes as is shown by Figures A-33a, A-34a, and A-35a.

The performance of the codes when quantizing sinusoids of amplitude 5 and periods ranging from 5 to 50 is as shown by Figures A-29 and A-36 thru A-40. The code rate and area error performance of the (1,3) and (1,2,3) codes were extremely close when quantizing the sinusoid of period 5 and amplitude 5 as shown by Figure A-36a. The (1) code had the worst performance for both code rate and area error. The performances of the (1,3) and (1,2,3) codes again are fairly close when quantizing the sinusoid of period 10 and amplitude of 5 with the (1,2,3) code being the best as is depicted by Figure A-37. The code rate for the (1) code dropped considerably when the period was increased from 5 to 10 as can by seen in Figures A-36a and A-37a. The code rate of the (1,3), (1,2,3) and (1,2) codes did not exhibit any appreciable change for the change in period from 5 to 10. The area error performance of the (1,3) and (1,2,3) codes is approximately the same when quantizing the sinusoid of period 20 and amplitude of 5 as is shown by Figure A-29b. There is a considerable difference in the code rates of the (1,3) and (1,2,3) codes as is depicted by Figure A-29a. The (1) code has the best code rate and the worst area error of all the codes as is shown by Figure A-29. When the period of the

sinusoid was increased to 30 the area error of the (1,3) and (1,2,3) codes was again approximately equal with the (1,2,3) code being the best as is shown by Figure A-38b. The code rate of the (1,3) and (1,2,3) codes were also approximately equal with the (1,3) code having the lowest code rate as is shown by Figure A-38a. The (1) code had the highest code rate and area error of the codes when the period of the sinusoid was increased from 30 to 50 as is shown by Figures A-38 thru A-40. The (1,3) code performed by far the best when quantizing the sinusoid of period 40 and amplitude of 5 as is shown by Figure A-39 . The performances of the (1,2) and (1,2,3) codes were very close together and in the center of the difference between the (1,3) and (1) codes performances as is shown by Figure A-39 . When the period was increased to 50 the (1,2,3) code had the lowest area error followed closely by the (1,3) code as is depicted by Figure A-40b. The (1,3) code had the lowest code followed by the (1,2,3) code as is shown by Figure A-40a.

Figures A-41 and A-42 depict the performances of the codes as the amplitude to period ratio is varied. In Figure A-41 the period is 10 and the amplitude ranges from 2 to 100. In Figure A-42 the amplitude is 5 and the period ranges from 20 to 100. The area error performances of the (1,2,3) and (1,3) codes are approximately equal as the amplitude to period ratio approaches 10 as is shown by Figure A-41b. The area error for all of the codes seems to be upper bounded by .25 as

was discussed earlier when analyzing the codes separately. The code rate performances of the codes as depicted by Figure A-41a was surprising. It was anticipated that as the amplitude to period ratio increased that the code rates of the multiple ringed codes would decrease instead of increase as they did. The reason for this performance is explained by Figure V-1 below.
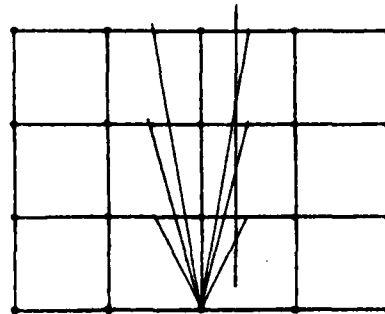
Figure V-1   Quantization of a Vertical Straight Line

As is shown by Figure V-1 above the sinusoids of increasing amplitude were not able to be efficiently quantized by the multiple ring codes since the link gate sets are almost never satisfied for the higher order rings. The code rate of the higher order codes exhibited a decrease as the amplitude to period ratio increased from .2 to 1.0 and then the code rate began to rise rapidly as is shown by Figure A-41a . The code rate of the (1) code exhibited an increase to its maximum as the ratio of amplitude to period increased from .2 to 2.0. As Figures A-41 and A-42 depict the performances of the higher order codes is very unpredictable for amplitude to period ratios under 1.0. The codes exhibit large flucuations in

performance for small changes in the amplitude to period ratio. The (1) code exhibts a decrease in code rate as the amplitude to period ratio is increased from .05 to .20 and then begins to increase as the amplitude to period ratio increases as is shown by Figures A-41a and A-42a. The area error of the (1) code decreases as the amplitude to period ratio increases from .05 to .25 and then increases as the amplitude to period ratio increases as is shown by Figures A-41b and A-42b .

Figure A-43 and A-44 depict the performance of the codes if the code rate and area error are given equal weighting for selection of the best code. Figure A-43 depicts the performance of the codes for amplitude to period ratios of .05 to .25 and Figure A-44 depicts the performance of the codes for amplitude to period ratios of .2 to 10.0. The performance of the (1,3) code is superior when the amplitude to period ratio is less than 4.0 as shown by Figures A-43 and A-44 . The (1,3) code is not very stable in this region but is consistiently the best performer. As the amplitude to period ratio increases above 4.0 the (1) code becomes the best performer as is shown by Figure A-44 .

1/2 Grid Square Capture Region

The enlarging of the capture region of the codes to 1/2 of a grid square did not affect the performance of the codes. This result was used in designing the algorithm for quantizing the rotated sinusoids.

Rotated Sinusoid Algorithm

The performance of the rotated sinusoid algorithm was minimal at best. The use of the La Grange Polynomial to approximate the line drawing is too inefficient with respect to computer processing time. The algorithm is also very dependent upon the number of points chosen to generate the coefficients of the La Grange Polynomial due to the number of operations required and accuracy of the computer. An additonal problem is that the particular value of X for which the polynomial is being evaluated must lie within the set of points selected to generate the La Grange Polynomial coefficients. If the value of X is outside of the interval of the points selected then extrapolation will be performed and the accuracy will be questionable. In an attempt to overcome this a test was performed to ensure that the value of the first point selected to generate the La Grange Polynomial coefficients was less than the value of X for which the polynomial was to be evaluated. An additional test should be performed to ensure that the value of X is not greater than the last point of the interval. When 5 points are used the processing time is much faster and the accuracy seems better than when 10 points are used. The intersect point finding subroutines seemed to work very well but in some cases the same intersect point is chosen twice in succession which causes problems for the La Grange Polynomial routine. This problem occurrs since then the points are no longer distinct and a divide by zero error will occur. A possible solution is

to reduce the step size from .2 to .1 when calculating the points to be rotated, however this will add to the already extremely high processing time.

Summary

In this chapter the performances of the (1), (1,2), (1,3), and (1,2,3) codes have been analyzed when quantizing sinusoids of different periods and amplitudes. The performance of the codes have been analyzed with respect to there dependence upon the amplitude to period ratio. The effect of changing the capture region of the quantizing code and the performance of the algorithm designed to quantize rotated sinusoids has been analyzed.

Chapter VI

Conclusions and Recommendations

In this chapter the conclusions drawn from the analysis of the performance of the chain codes will be discussed and recommendations for future studies will be made.

Conclusions

From the analysis of chapter V it can be concluded that the amplitude to period ratio is not a very good predictor of how well a code will perform. A better predictor might be the ratio of the grid size to the amplitude or the period of the line drawing. The position of the point of zero slope of the sinusoids with respect to the horizontal grid lines of the quantizing grid had a very definite impact upon the performance of the code. The size of the capture region does not affect the performance of the codes.

Recommendations for Future Study

The following recommendations are made for future study in this area:

1. Develop a general algorithm capable of approximating the X and Y intersects of any function, whether it be single or multivalued.

2. Implement the parallel quantizing scheme.

3. Study the effect of the position of zero slope of a sinusoid with respect to the horizontal grid lines of the quantizing grid.

4. Continue the implementation of the algorithm to quantize rotated sinusoids.

5. Quantize a function that does not have a constant peak amplitude, such as the sampling function.

6. Implement the algorithm so that the grid size can be varied with respect to the period and amplitude.

BIBLIOGRAPHY

1. Castor, K. G. and Neuhoff, D. L. "A Rate and Distortion Analysis for Grid Intersect Encoding of Line Drawings", _IEEE 1981 Pattern Recognition & Image Processing Proceedings_, IEEE Computer Society Conference on Pattern Recognition and Image Processing, Dallas Texas, August 1981.

2. Freeman, Herbert. _The Generalized Chain Code for Map Data Encoding and Processing._ Technical Report CRL-59. Air Force Office of Scientific Research. June 1978.

3. Freeman, Herbert. "Computer Processing of Line Drawing Images", _Computing Surveys, 6, No 1_: 57-97 (March 1974).

4. Freeman, Herbert and Glass, Jeremy M. "On The Quantization of Line Drawing Data", _IEEE Transaction Systems, Science, and Cybernetics SSC-5_:70-79 (January 1969).

5. Freeman, Herbert. "Some New Map Data Encoding Schemes", _Proceedings of the 3rd Jerusalem Conference on Information Technology._ Jerusalem, Israel. August 1978.

6. Freeman, Herbert. "Application of the Generalized Chain Coding Scheme to Map Data Processing", _Proceedings IEEE Computer Society Conference on Pattern Recognition and Image Processing._ Chicago, Ill. May 1978.

7. Freeman, Herbert and Saghri, A. "Generalized Chain Codes for Planar Curves", _Proceedings of the 4th International Joint Conference on Pattern Recognition._ Kyoto, Japan. November 1978.

8. Freeman, Herbert. "Efficient Representation of Spatial Data", _IEEE 1981 Pattern Recognition & Image Processing Proceedings_, IEEE Computer Society Conference on Pattern Recognition and Image Processing. Dallas,Texas. August 1981.

9. Glass, Jeremy. "A Criterion for the Quantization of Line Drawing Data" _Doctoral Dissertation_, New York University, June 1965.

10. Jones, Keith. "Grid-Based Line Drawing Quantization". _Master Thesis_ AFIT/GE/EE/82D-41. Air Force Institute of Technology, Wright Patterson Air Force Base, Ohio. December 1982.

11. Maxwell, P. C. "The Perception and Description of Line Drawings by Computer", Computer Graphics and Image Processing,1:31-46(April 1972).

12. Saghri, John A. Efficient Encoding of Line Drawing Data with Generalized Chain Codes, Tech Rept. IPL-TR-79-003, Image Processing Laboratory, Rensselaer Polytechnic Institute, Troy, N.Y., August 1979.

13. Freeman, H. and Frauenknecht, P. J. "An Efficient Computer Representation Scheme for Linear Map Data", IEE 1982 Pattern Recognition & Image Processing Proceedings.

14. Freeman, H. "Map and Line-Drawing Processing", in J. C. Simon and R. M. Havalick, Digital Image Processing, D. Reidel Publishing Company, Boston, Mass. 1981, pp 363-382.

15. Freeman, H. "Analysis of Line Drawings", in J. C. Simon and A. Rosenfeld, Digital Image Processing and Analysis, Noordhoff, Leyden, 1977, pp 187-199.

16. Freeman, H. and Saghri, J. A. "Comparative Analysis of Line-Drawing Modeling Schemes", Computer Graphics and Image Processing, vol 12, pp 203-223, 1980.

17. Freeman, H. "Computer Processing of Line-Drawings", Tech Rept. 403-30, Department of Electrical Engineering and Computer Science, New York University, Bronx, New York, May 1973.

18. "Panel on Map Data Processing, Chairman: Freeman, Herbert, IEEE 1981 Pattern Recognition & Image Processing Proceedings, IEEE Computer Society Conference on Pattern Recognition and Image Processing, Dallas, Texas, August 1981.

19. Castor, K. G. and Neuhoff, D. L., "A Rate and Distortion Analysis for Grid Intersect Quantization of Line Drawing Images." Proceedings Eighteenth Annual Allerton Conference on Communcation, Control, and Computing, University of Illinois at Urbana-Champaign, October 1980.

20. Santalo, Luis A., Encyclopedia of Mathematics and its Applications, Reading, Mass: Addison-Wesley Publishing Company, 1976.

21. Conte, S. D. and deBoor, C., Elementary Numerical Analysis: An Algorithmic Approach(2nd Ed),New York: McGraw-Hill Book Company, 1972.

22. Conte, S. D. and deBoor, C., <u>Elementary Numerical Analysis(3rd Ed)</u>, New York: McGraw-Hill Book Company, 1980.

23. Vandergraft, James S., <u>Introduction to Numerical Computations</u>, New York: Academic Press, 1978.

24. McCalla, Thomas R., <u>Introduction to Numerical Methods and Fortran Programing</u>, New York: John Wiley and Sons Inc., 1976.

25. Kreyszig, Erwin, <u>Advanced Engineering Mathematics(3rd Ed)</u>, New York: John Wiley and Sons Inc., 1972.

26. Szidarovszky, Fernic and Yakowitz,Sidney, <u>Principles and Procedures of Numerical Analysis</u>, New York: Plenum Press, 1978.

Appendix A

Performance Plots

In each figure, the upper plot is that of the accumulated encoding rate per unit length versus the domain of the function. The lower plot is that of the accumulated area error per unit length versus the domain of the function. Each performance curve in the figures is annotated with either the amplitude of the sinusoid being quantized, or the code which was used to do the quantization. Table A-1 depicts the asymptotic values of the area error and code rate of the codes when quantizing sinusoids of periods 10 and 20 with amplitudes of 500 and 1000. Tables A-2 and A-3 depict the asymptotic performance of the (1) and (1,3) codes quantizing sinusoids of period 20 and amplitudes ranging from 10.0 to 11.0 and 20.0 to 21.0.

APPENDIX A

LIST OF FIGURES

APPENDIX A

LIST OF TABLES

Figure A-1a

Figure A-1b

AMPLITUDES

2   4   6   8   10

A-5

Figure A-2a

BPL

3.20
3.00
2.80
2.60

20
18
16
14
12

0.00   40.00   80.00   120.00   160.00   200.00

X/50

(1)   CODE   PHASE .0   PER = 10

AMPLITUDES

12 14 16 18 20

Figure A-2b

AEPL

0.24
0.22
0.20
0.18

20
18
16
14
12

0.00   40.00   80.00   120.00   160.00   200.00

X/50

A-6

Figure A-3a

Figure A-3b

Figure A-4a

(1)   CODE   PHASE . 0   PER - 10

AMPLITUDES

60 70 80 90 100

Figure A-4b

A-8

Figure A-5a

(1)   CODE   PHASE . 0   PER = 20

AMPLITUDES

5. 1 5. 2 5. 3 5. 4 5. 5

Figure A-5b

A-9

Figure A-6a

< 1 >   CODE   PHASE .0   PER = 20

AMPLITUDES

5.6 5.7 5.8 5.9 6.0

Figure A-6b

A-10

Figure A-7a

Figure A-7b

Figure A-8a

Figure A-8b
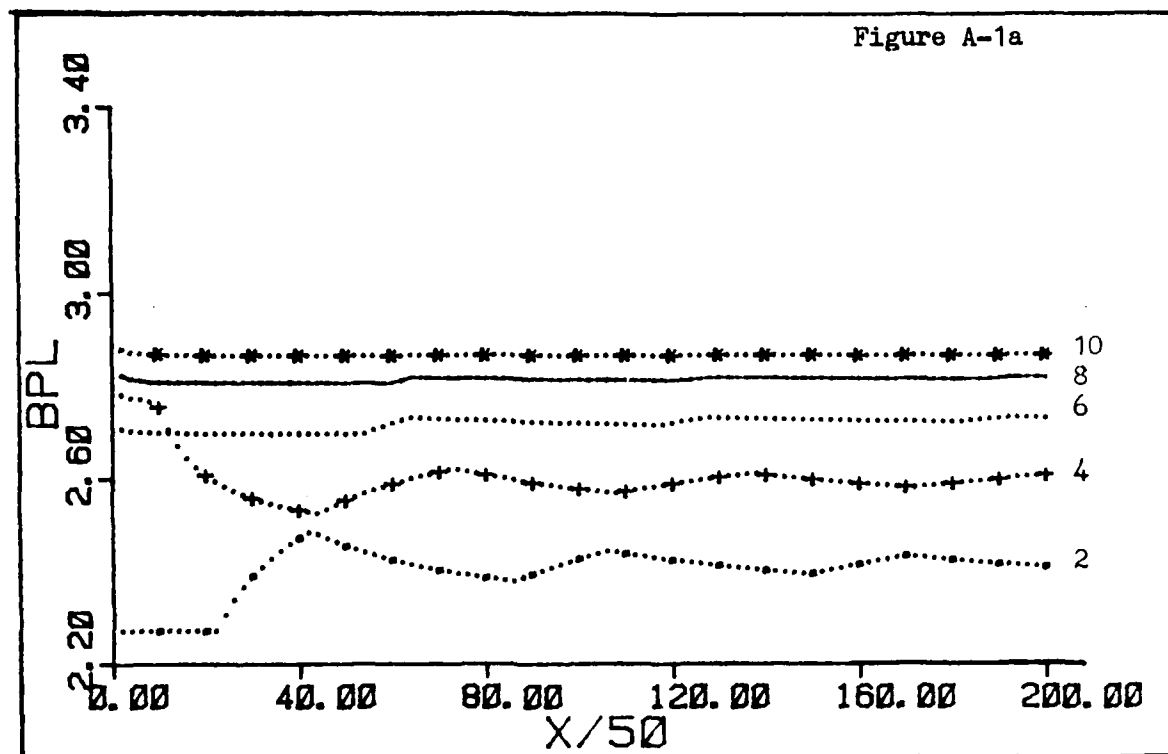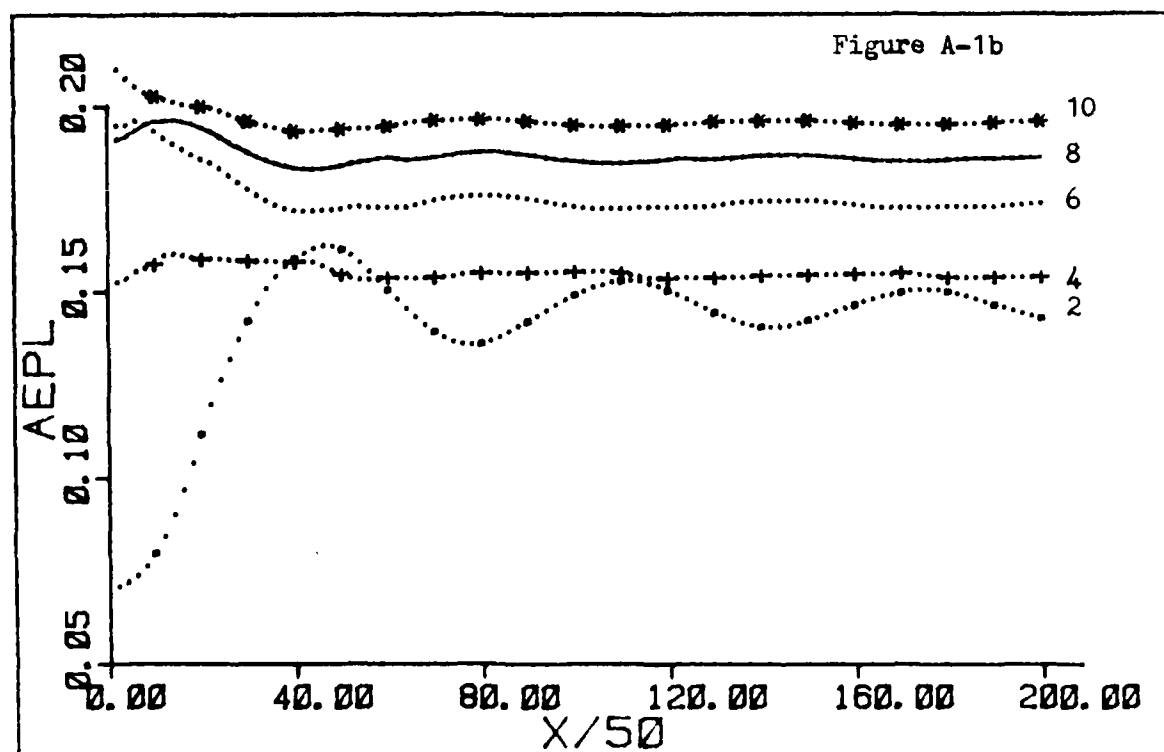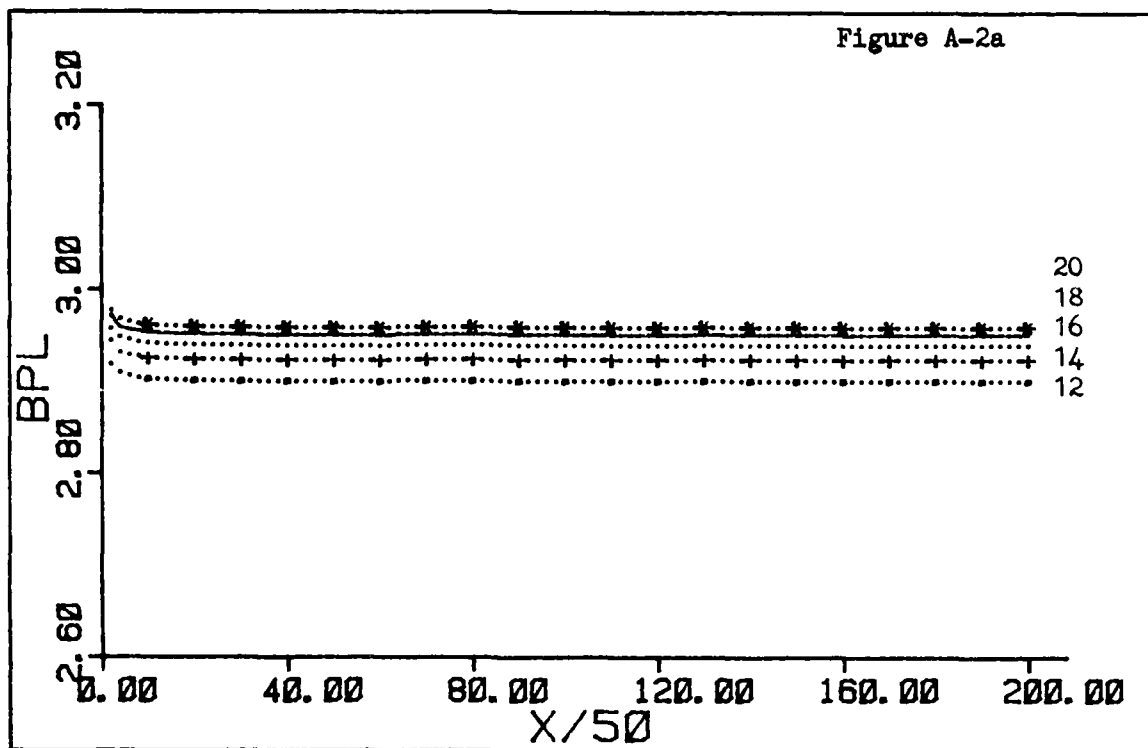
AMPLITUDES

2   4   6   8   10

(1.2)   CODE   PHASE .0   PER = 10

A-12

Figure A-9a

<1.2>   CODE   PHASE .0   PER = 10

AMPLITUDES

12 14 16 18 20

Figure A-9b

A-13

Figure A-10a

BPL

X/50

(1,2)   CODE   PHASE .0   PER = 10

AMPLITUDES

10 20 30 40 50

Figure A-10b

AEPL

X/50

A-14

Figure A-11a

Figure A-11b

AMPLITUDES

60 70 80 90 100

(1,2)   CODE   PHASE .0   PER = 10

A-15

Figure A-12a

(1, 2)   CODE   PHASE . 0   PER = 20

AMPLITUDES

5. 1 5. 2 5. 3 5. 4 5. 5

Figure A-12b

Figure A-13a

Figure A-13b

AMPLITUDES

5.6 5.7 5.8 5.9 6.0

A-17

Figure A-14a

(1.2)   CODE   PHASE .0   PER = 20

AMPLITUDES

5   10 20 40 60

Figure A-14b

Figure A-15a

<1.3>   CODE   PHASE .0   PER ≈ 10

AMPLITUDES

2 4 6 8 10

Figure A-15b

A-1

END

MICROCOPY RESOLUTION TEST CHART

NATIONAL BUREAU OF STANDARDS-1963-A

Figure A-16a

<1.3>   CODE   PHASE .0   PER - 10

AMPLITUDES

12 14 16 18 20

Figure A-16b

A-20

Figure A-17a

(1.3)   CODE   PHASE .0   PER — 10

AMPLITUDES

10 20 30 40 50

Figure A-17b

A-21

Figure A-18a

<1.3> CODE PHASE .0 PER - 10

AMPLITUDES

60 70 80 90 100

Figure A-18b

A-22

Figure A-19a

(1, 3)   CODE   PHASE . 0   PER = 20

AMPLITUDES

5. 1 5. 2 5. 3 5. 4 5. 5

Figure A-19b

Figure A-20a

BPL

X/100

(1.2)   CODE   PHASE . 0   PER = 20

AMPLITUDES

5.6 5.7 5.8 5.9 6.0

Figure A-20b

AEPL

X/100

A-24

Figure A-21a

Figure A-21b

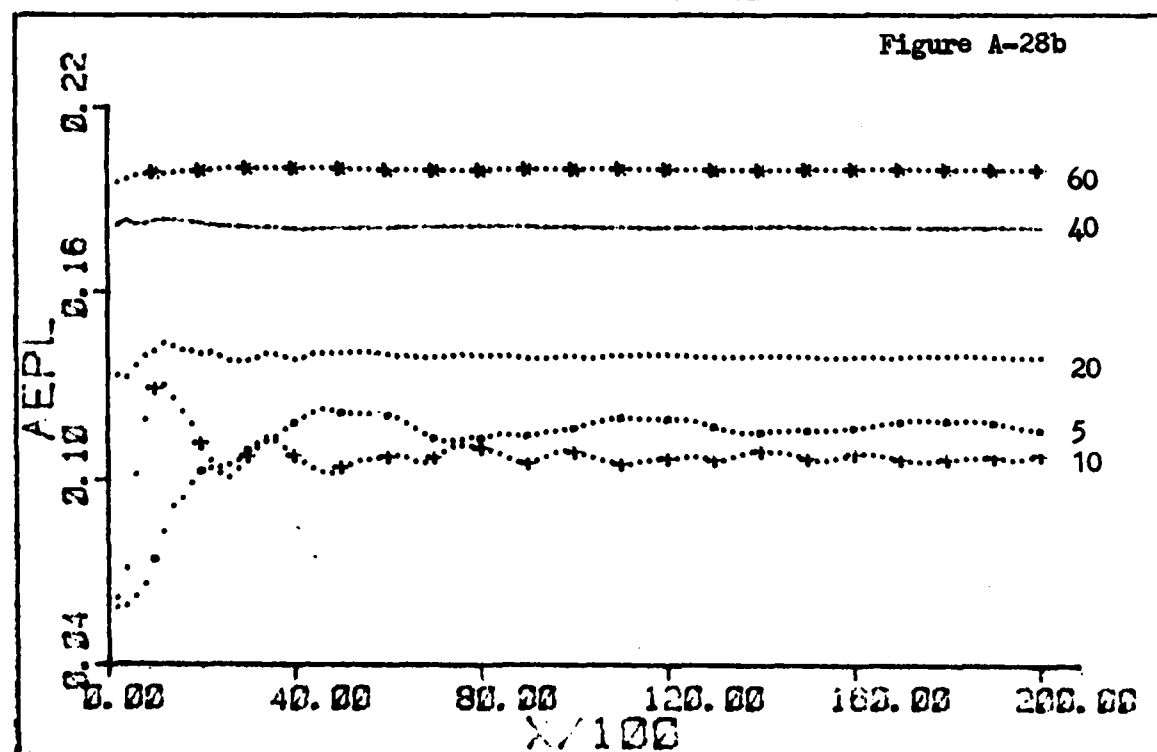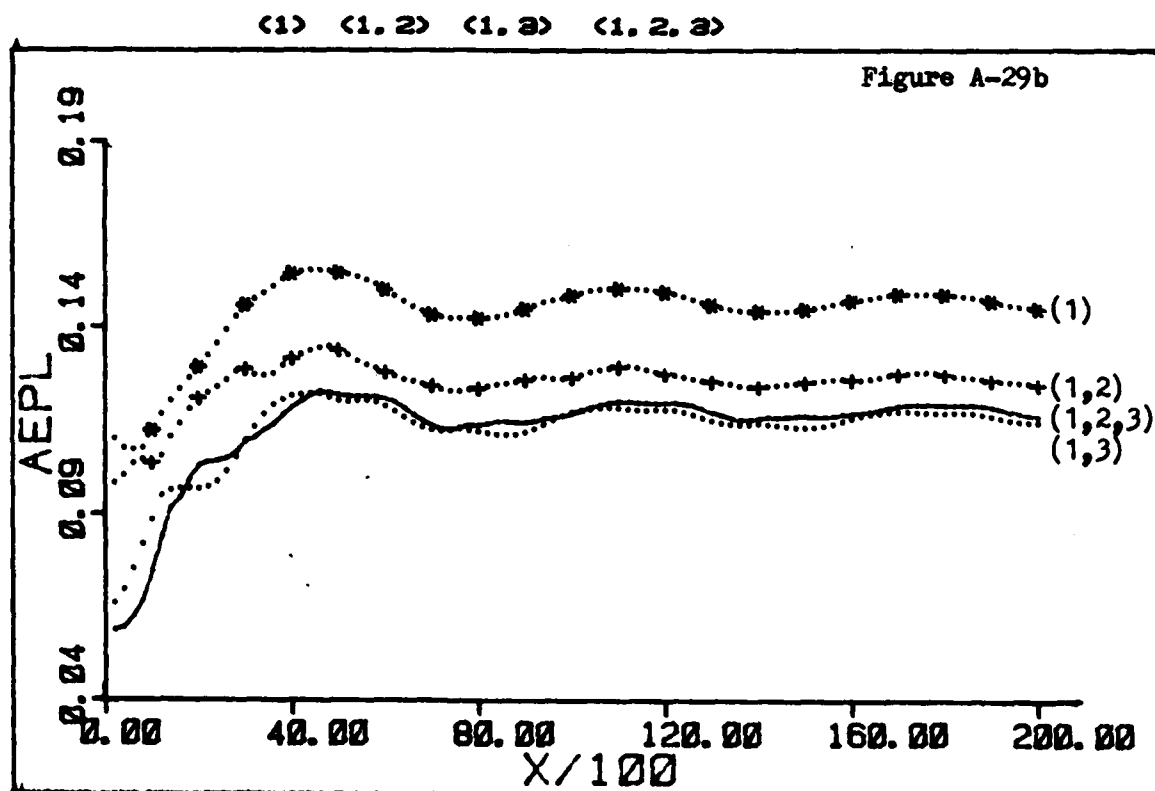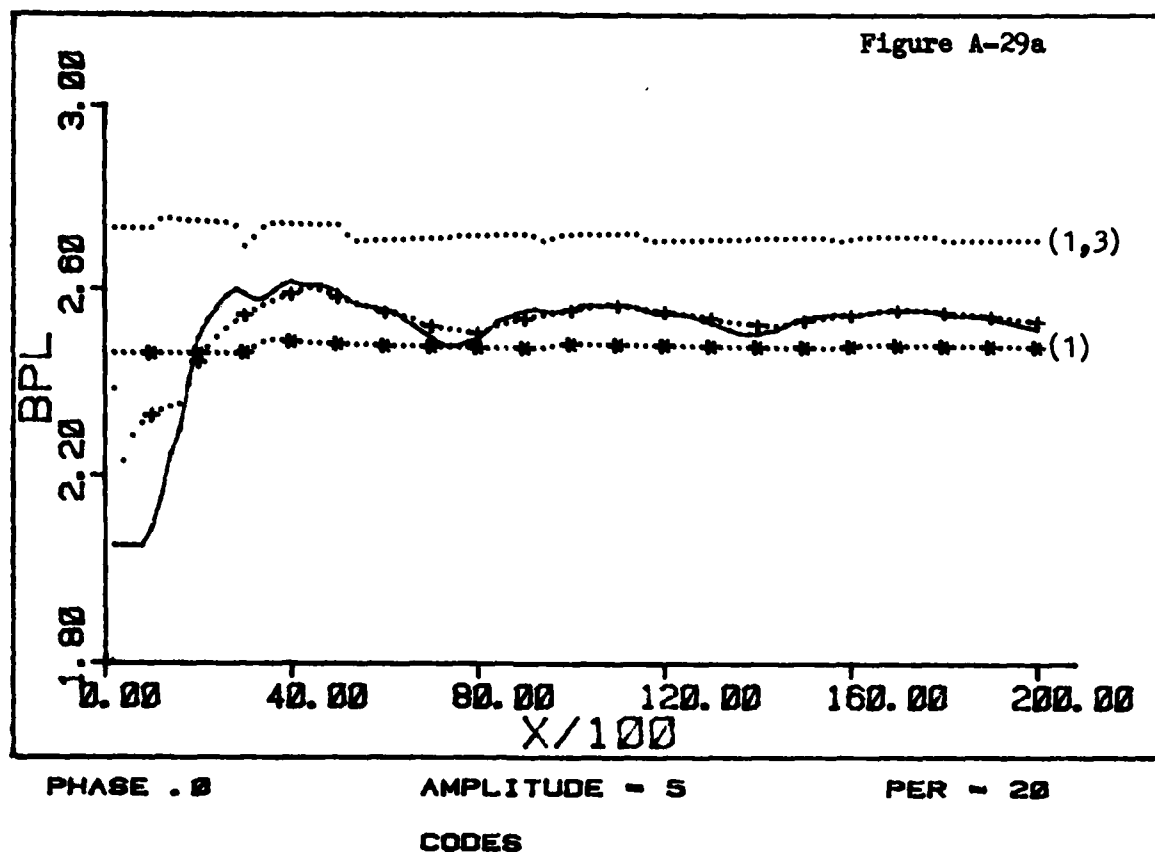<1.3> CODE   PHASE .0   PER = 20

AMPLITUDES

5 10 20 40 60

A-25

Figure A-22a

<1, 2, 3> CODE   PHASE . 0   PER - 10

AMPLITUDES

2   4   6   8   10

Figure A-22b

A-26

Figure A-23a

(1, 2, 3) CODE   PHASE . 0   PER = 10

AMPLITUDES

12 14 16 18 20

Figure A-23b

A-27

Figure A-24a

<1. 2. 3> CODE   PHASE . 0   PER — 10

AMPLITUDES

10 20 30 40 50

Figure A-24b

Figure A-25a

(1, 2, 3) CODE   PHASE . 0   PER = 10

AMPLITUDES

60 70 80 90 100

Figure A-25b

A-29

Figure A-26a

BPL

3.70
3.20
2.70
2.20

5.5
5.4
5.3
5.1
5.2

0.00   40.00   80.00   120.00   160.00   200.00

X/100

(1, 2, 3) CODE   PHASE .0   PER = 20

AMPLITUDES

5.1 5.2 5.3 5.4 5.5

Figure A-26b

AEPL

0.22
0.16
0.10
0.04

5.5
5.4
5.3
5.2
5.1

0.00   40.00   80.00   120.00   160.00   200.00

X/100

A-30

Figure A-27a

Figure A-27b

(1, 2, 3) CODE   PHASE . 0   PER = 20

AMPLITUDES

5. 6 5. 7 5. 8 5. 9 6. 0

Figure A-28a

Figure A-28b

(1, 2, 3) CODE   PHASE . 0   PER ~ 20

AMPLITUDES

5   10 20 40 60

A-32

Figure A-29a

PHASE . 0   AMPLITUDE = 5   PER = 20

CODES

(1)   (1,2)   (1,3)   (1,2,3)

Figure A-29b

A-33

Figure A-30a

PHASE . 0   AMPLITUDE ▬10   PER ▬ 20

CODES

(1)   (1,2)   (1,3)   (1,2,3)

Figure A-30b

A-34

Figure A-31a

PHASE .0   AMPLITUDE -20   PER - 20

CODES

(1) (1,2) (1,3) (1,2,3)

Figure A-31b

A-35

Figure A-32a

PHASE . 0   AMPLITUDE =40   PER = 20

CODES

(1)   (1,2)   (1,3)   (1,2,3)

Figure A-32b

Figure A-33a

PHASE . 0   AMPLITUDE =60   PER = 20

CODES

(1) (1,2) (1,3) (1,2,3)

Figure A-33b

A-37

Figure A-34a

PHASE . 0   AMPLITUDE -80   PER - 20

CODES

(1) (1,2) (1,3) (1,2,3)

Figure A-34b

A-38

Figure A-35a
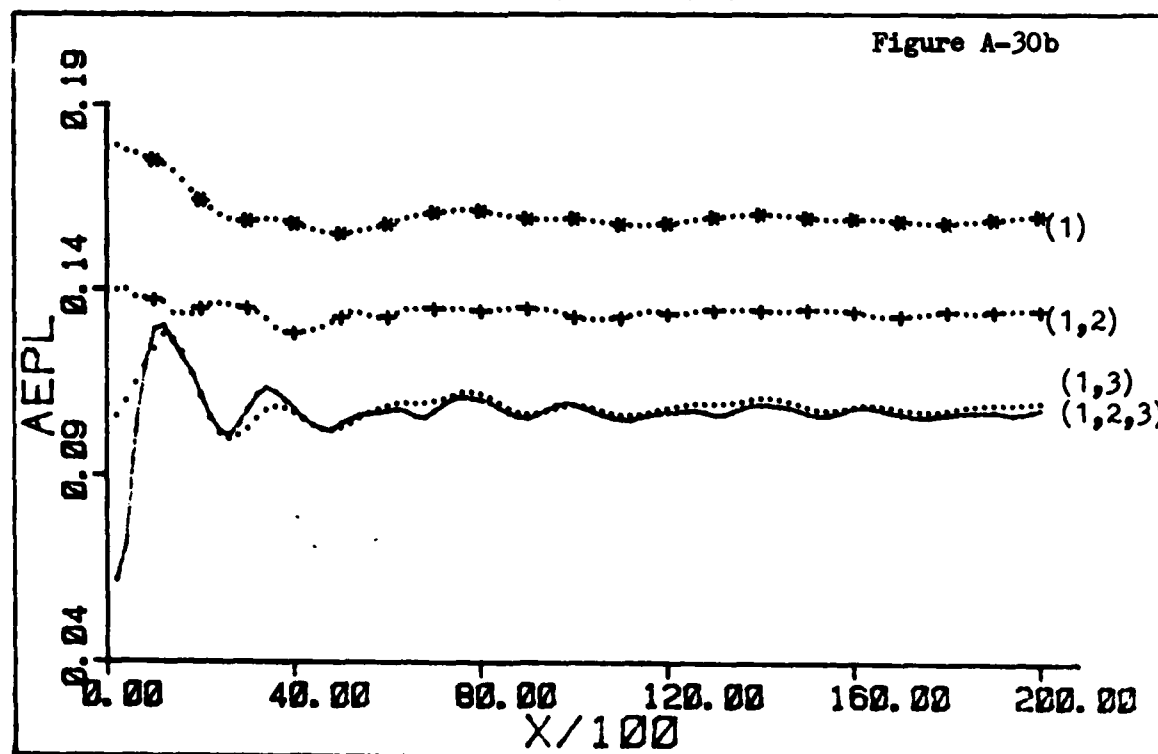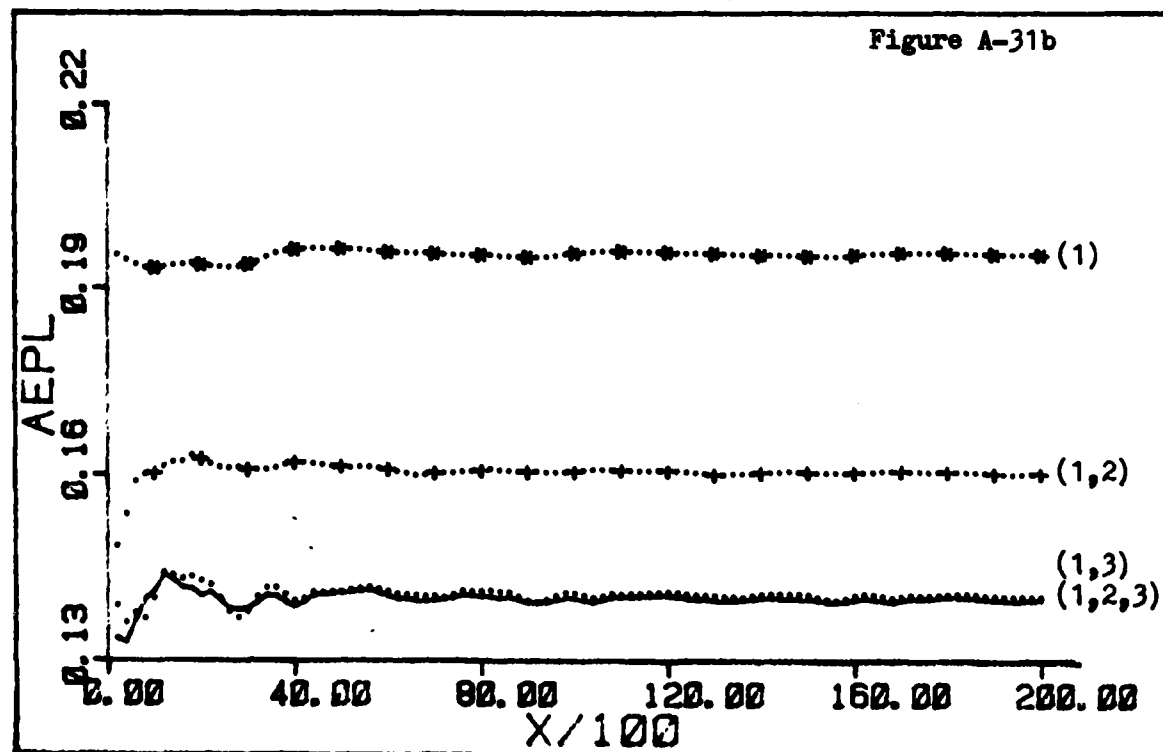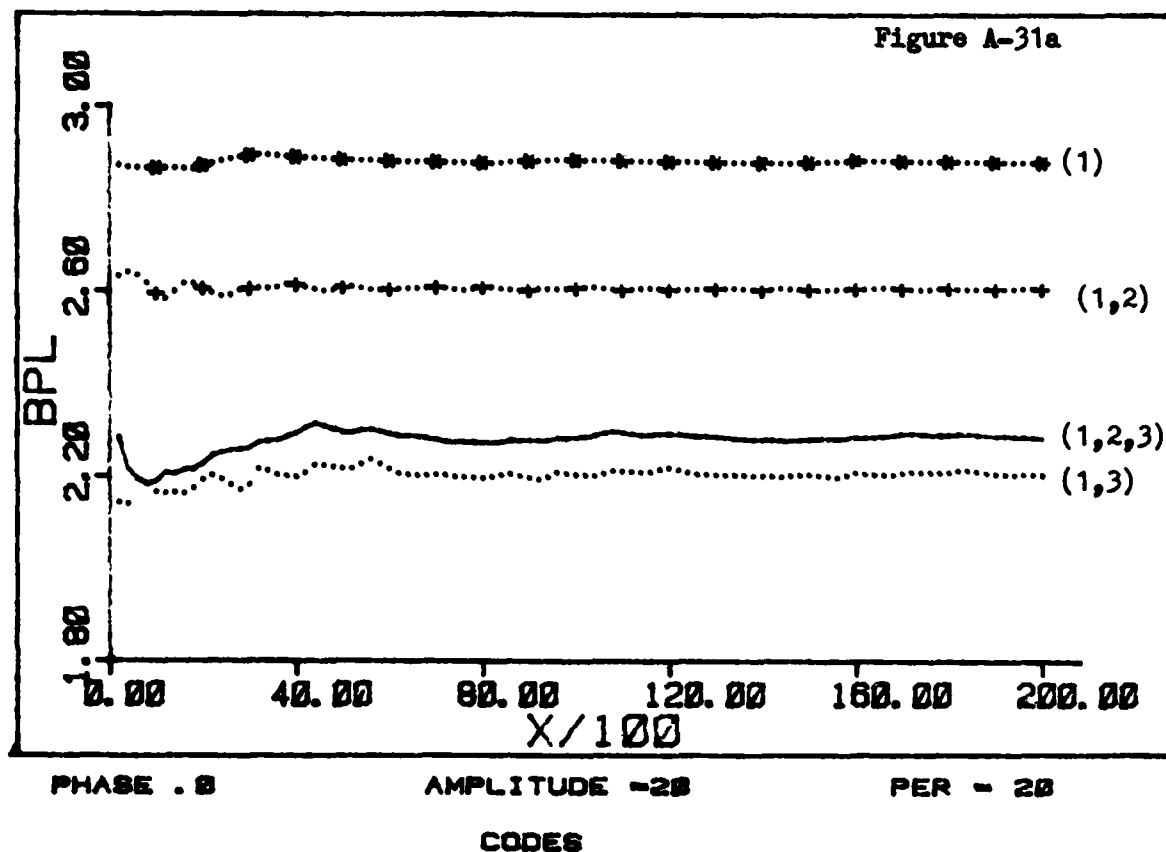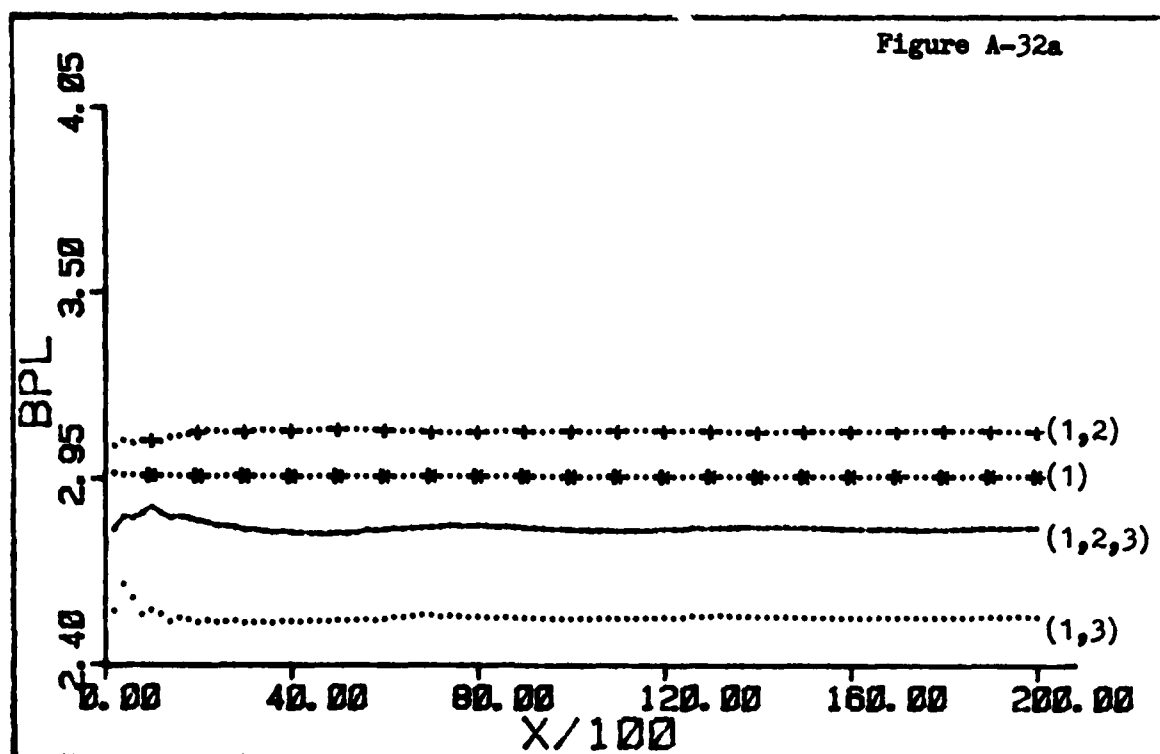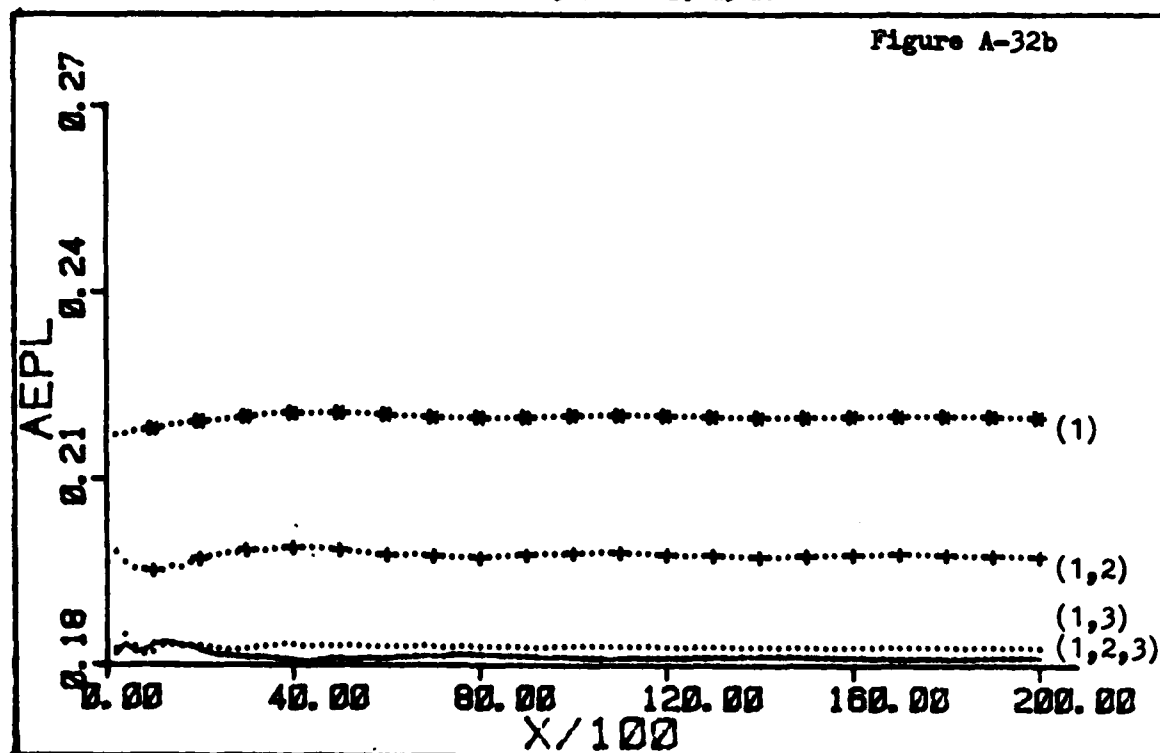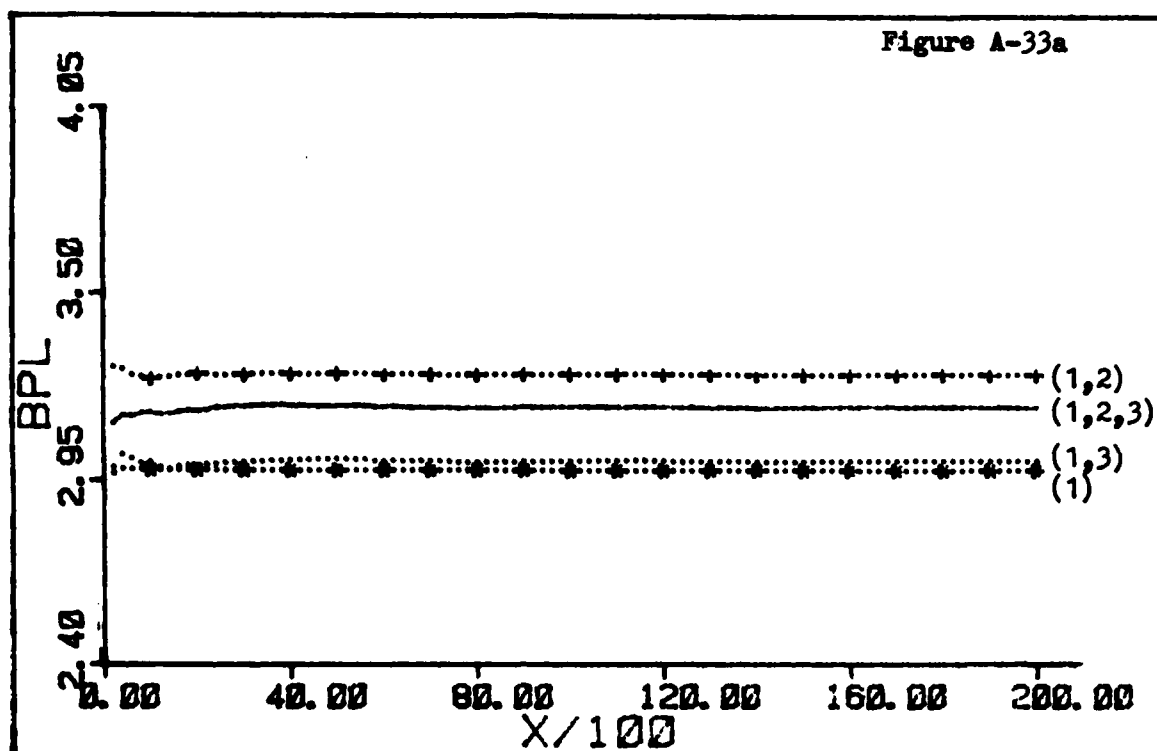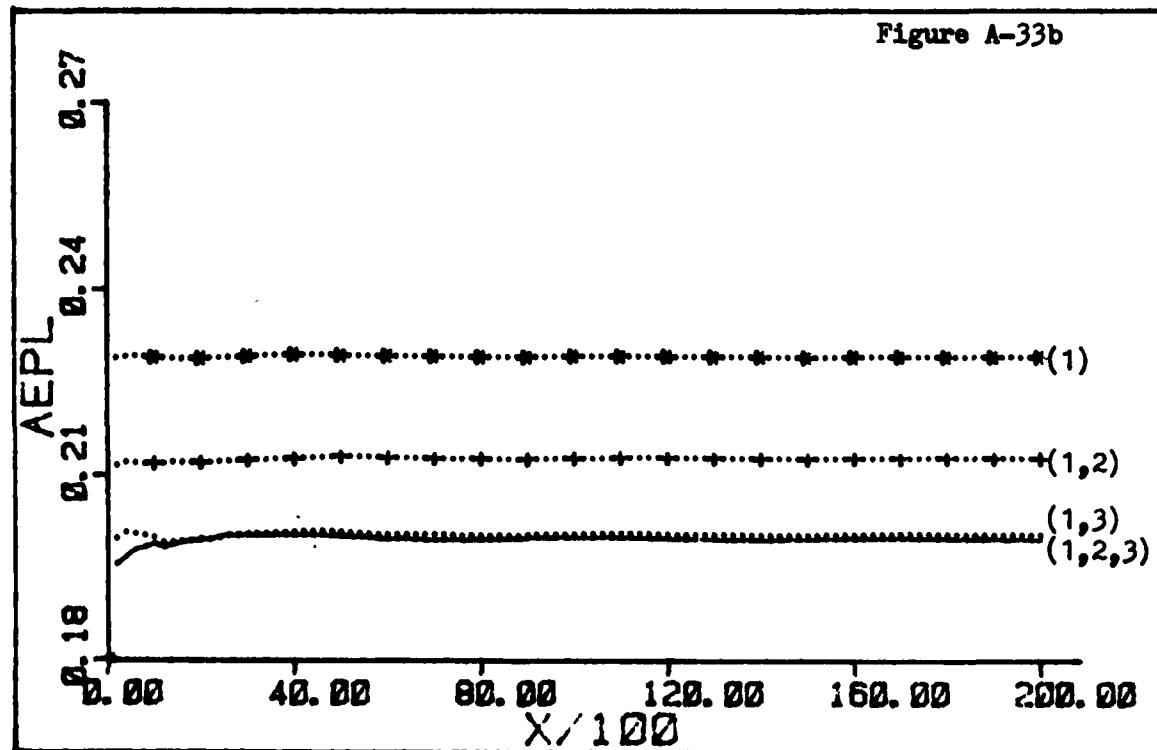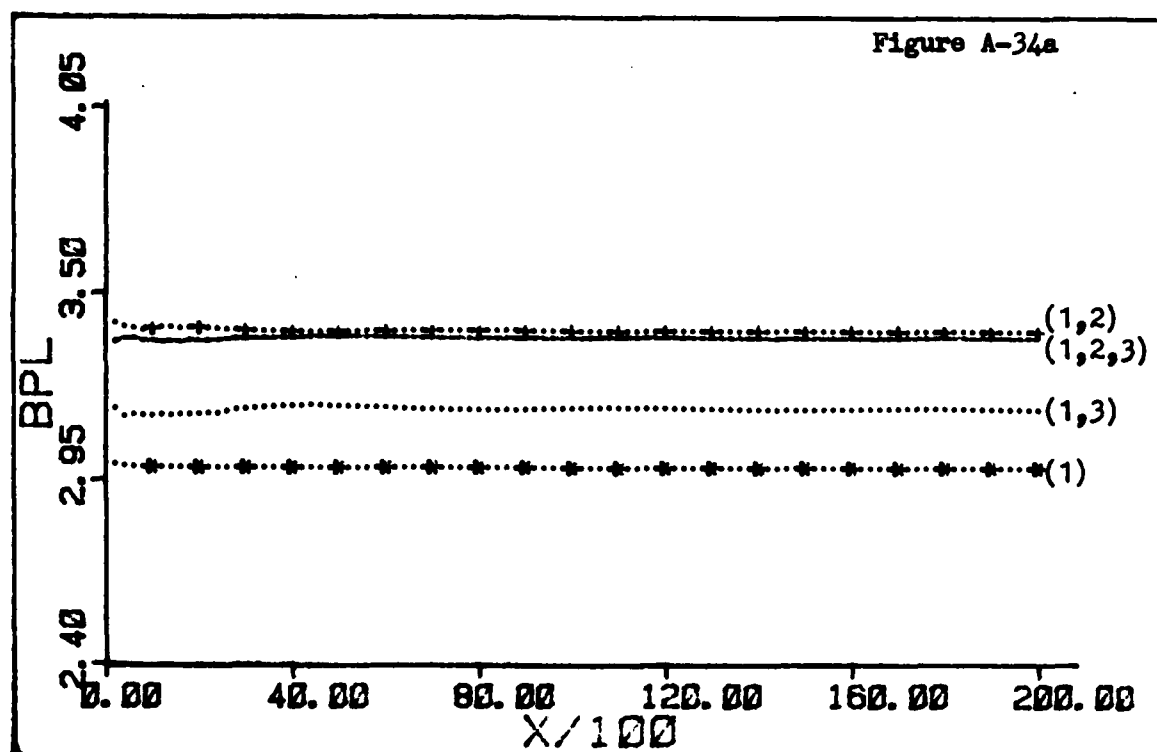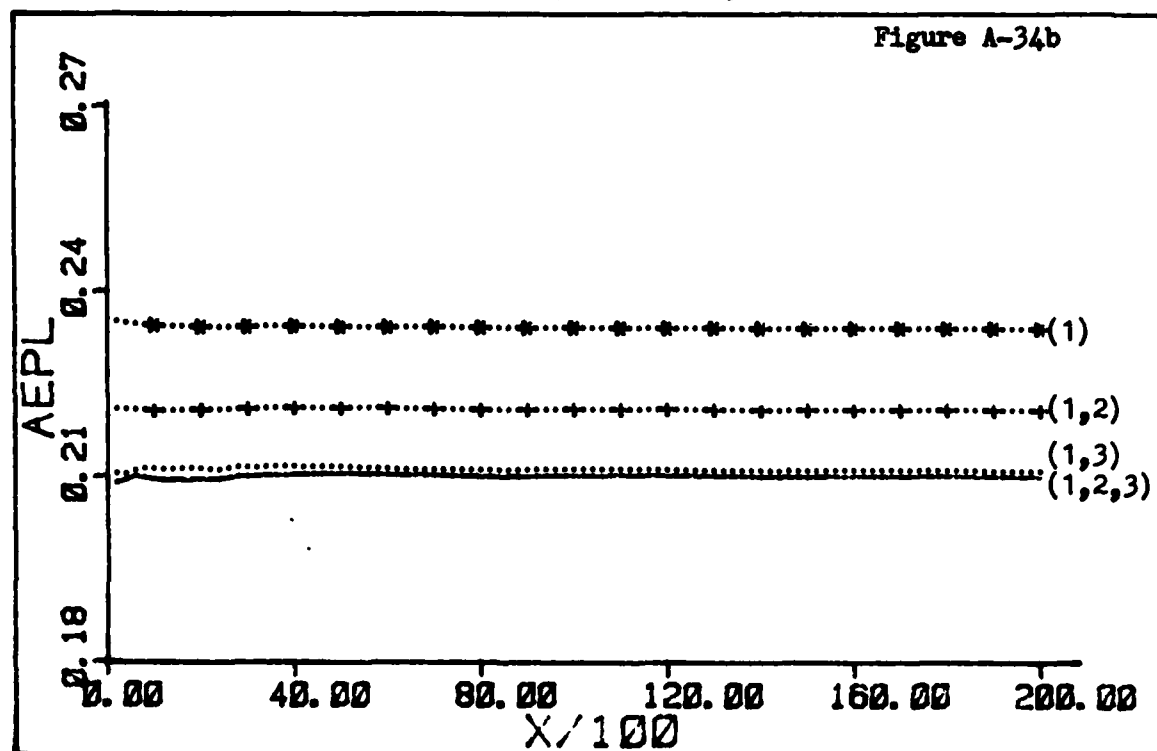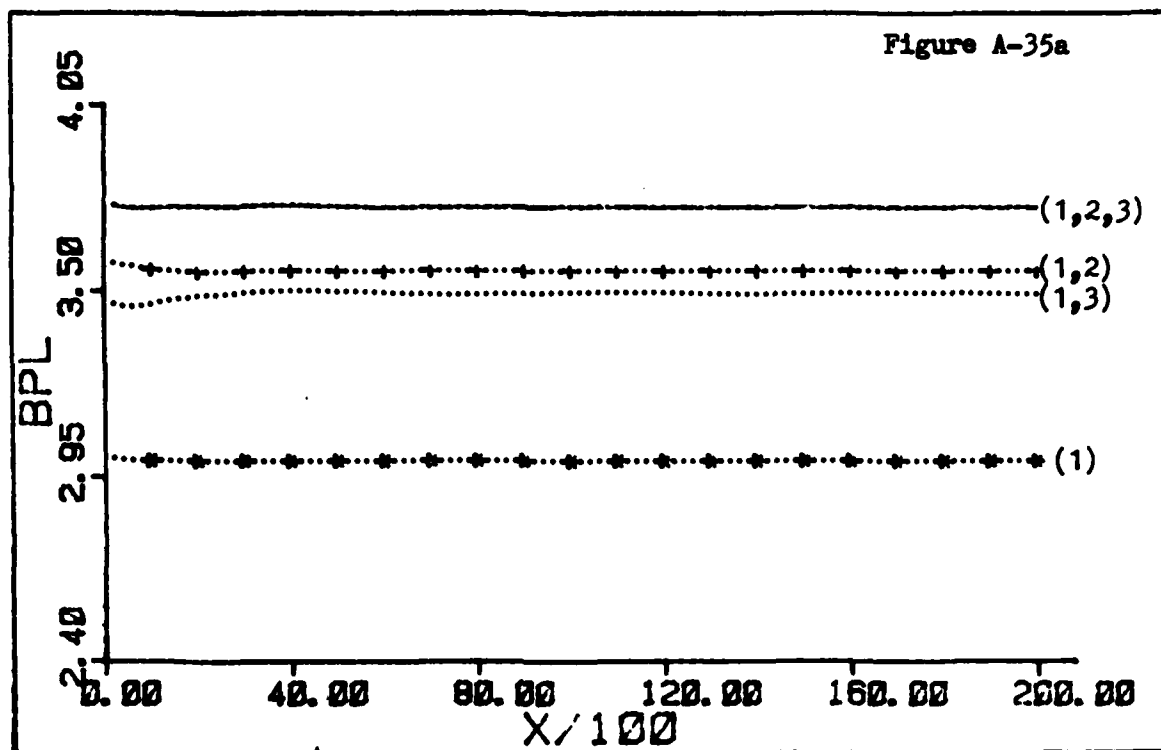
PHASE . 0   AMPLITUDE -100   PER - 20

CODES

(1)   (1,2)   (1,3)   (1,2,3)

Figure A-35b

A-39

Figure A-36a

PHASE . 0   AMPLITUDE = 5   PER = 5

CODES

(1)   (1, 2)   (1, 3)   (1, 2, 3)

Figure A-36b

A-40

Figure A-37a
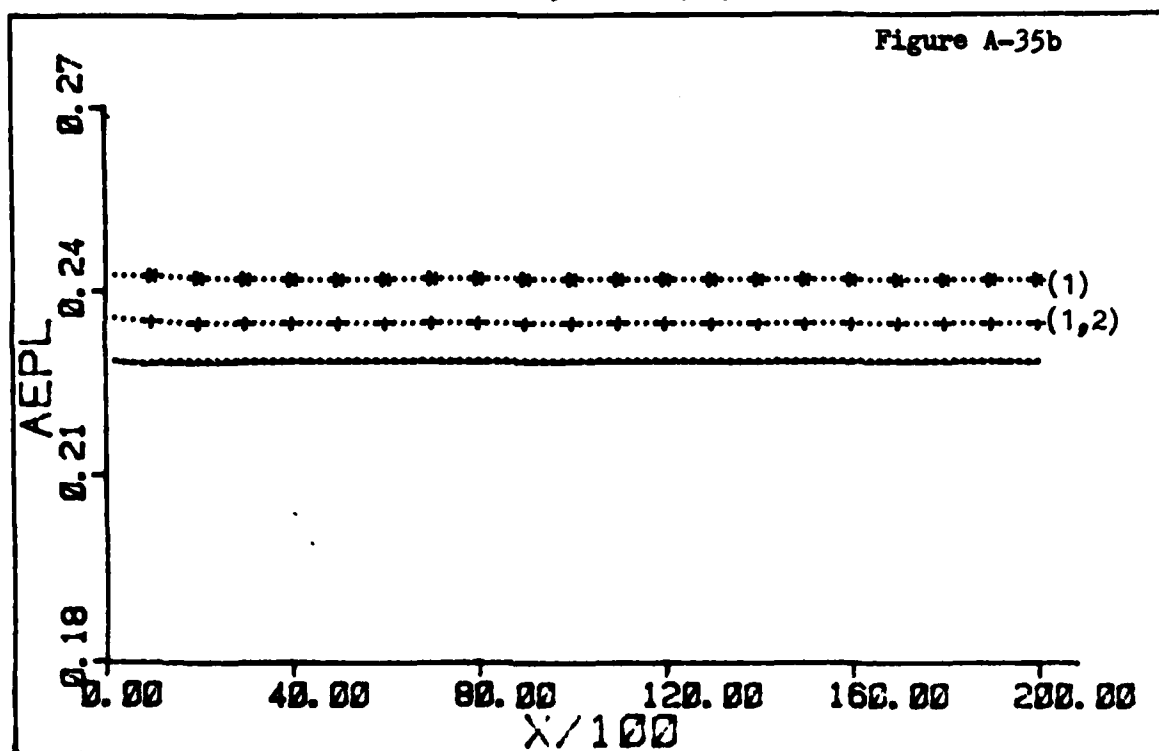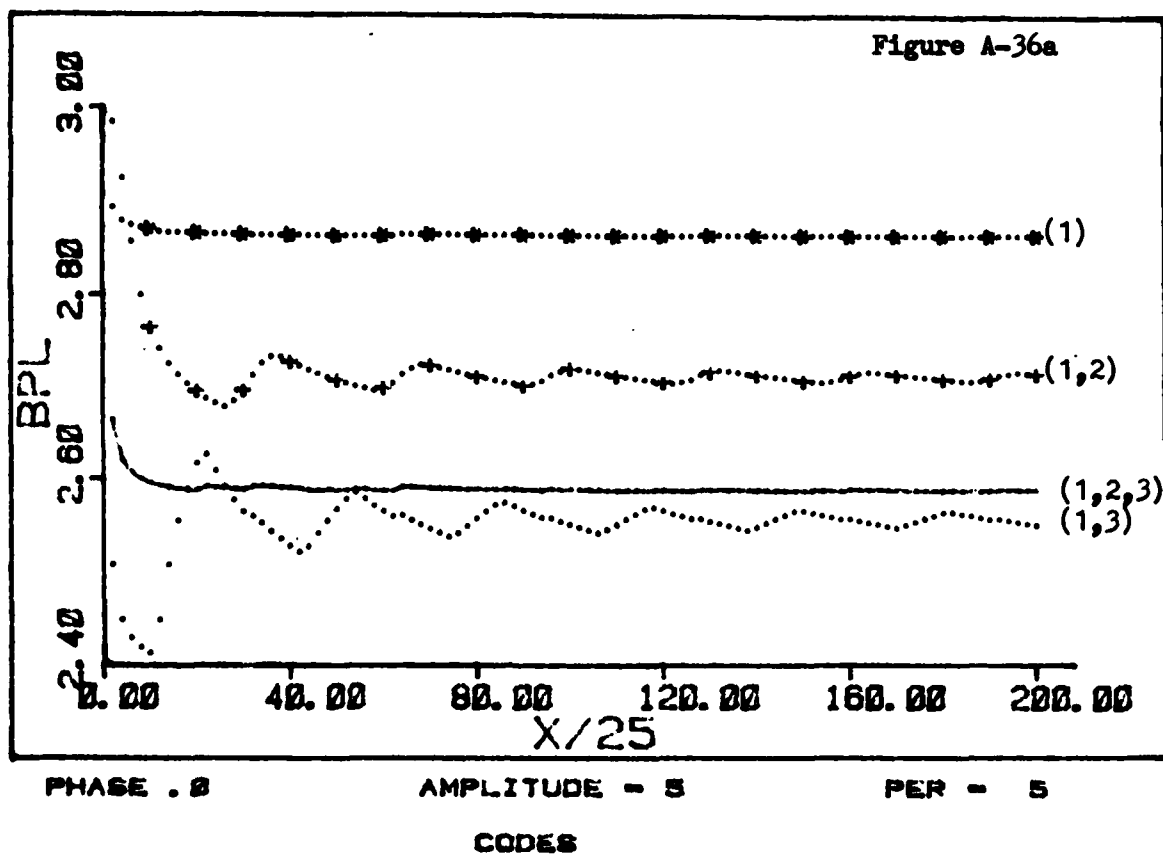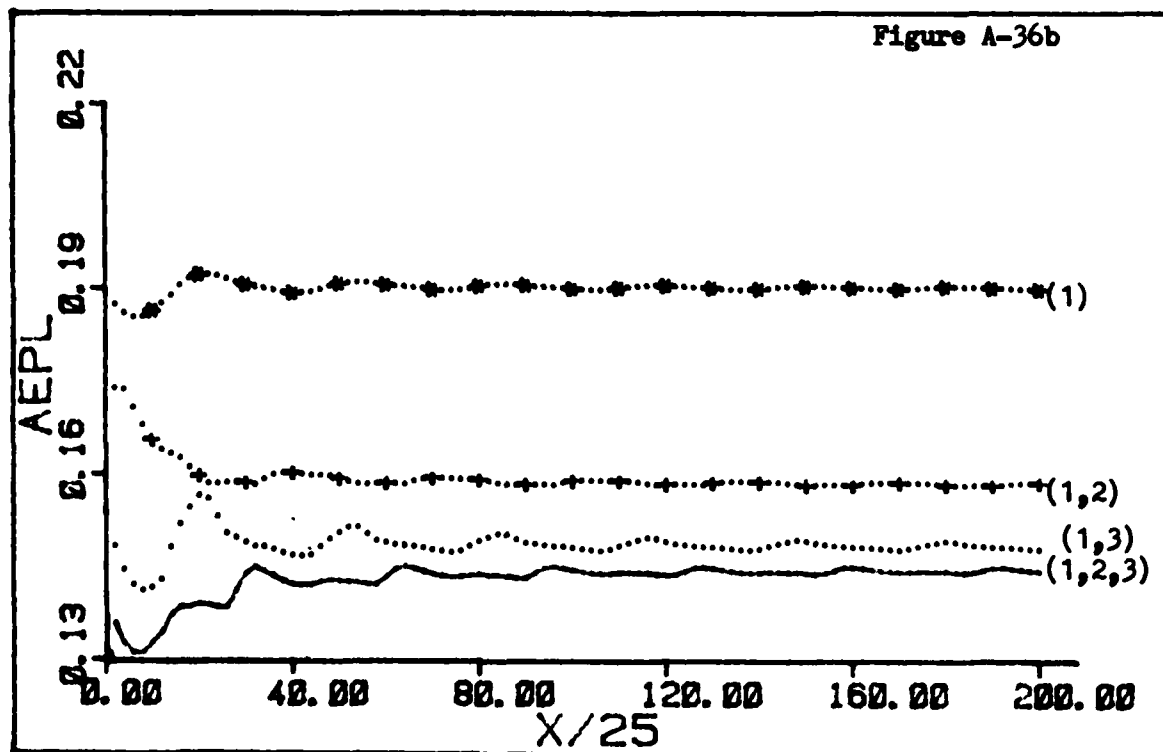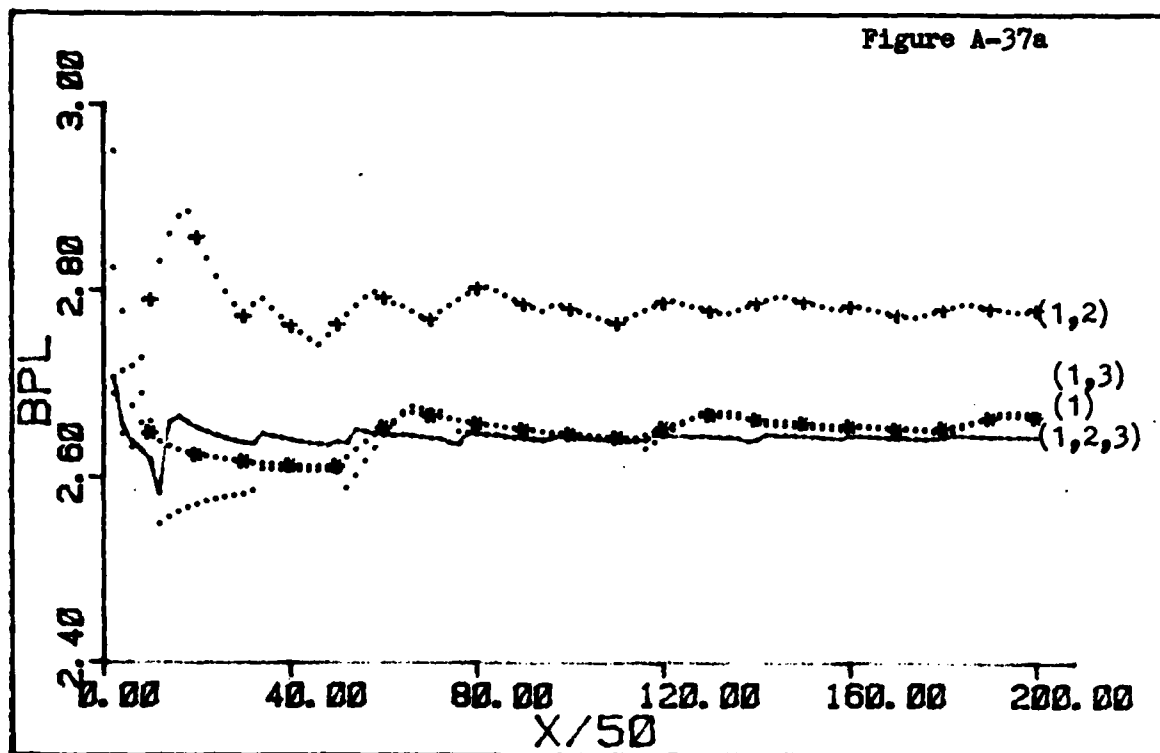
PHASE . 0   AMPLITUDE = 5   PER = 10

CODES

(1) (1, 2) (1, 3) (1, 2, 3)

Figure A-37b

Figure A-38a

PHASE . 0   AMPLITUDE = 5   PER = 90

CODES

(1) (1,2) (1,3) (1,2,3)

Figure A-38b

Figure A-39a

BPL

X/200
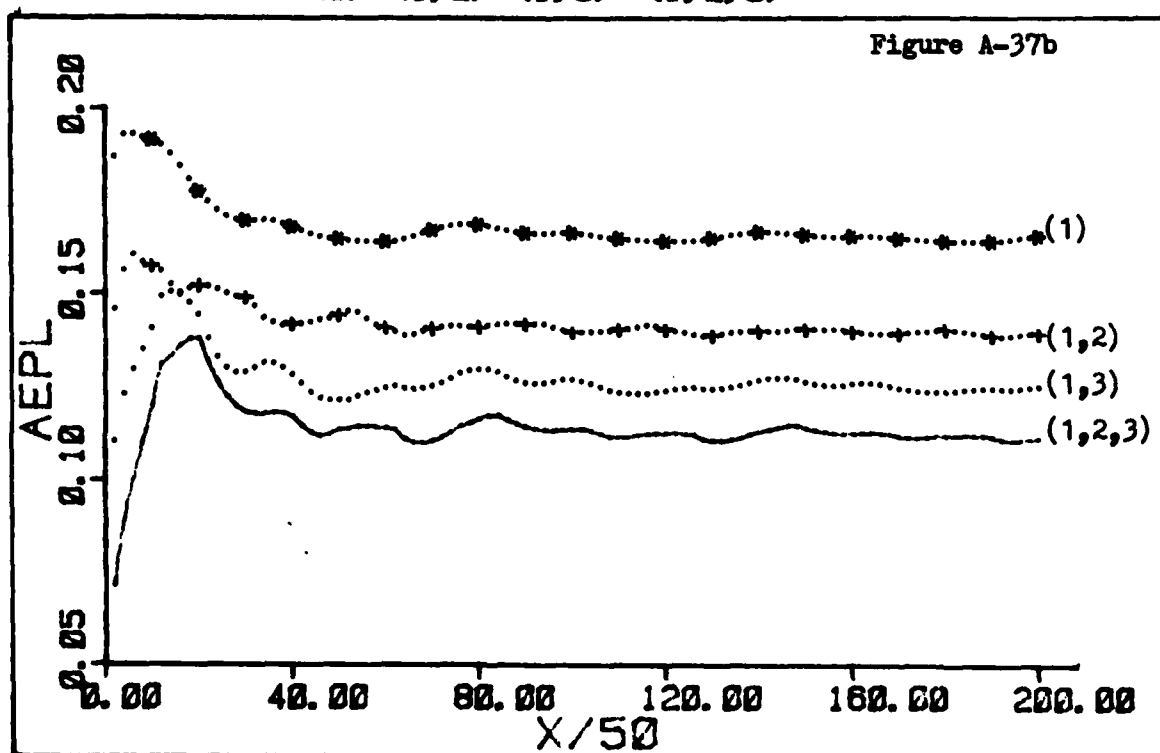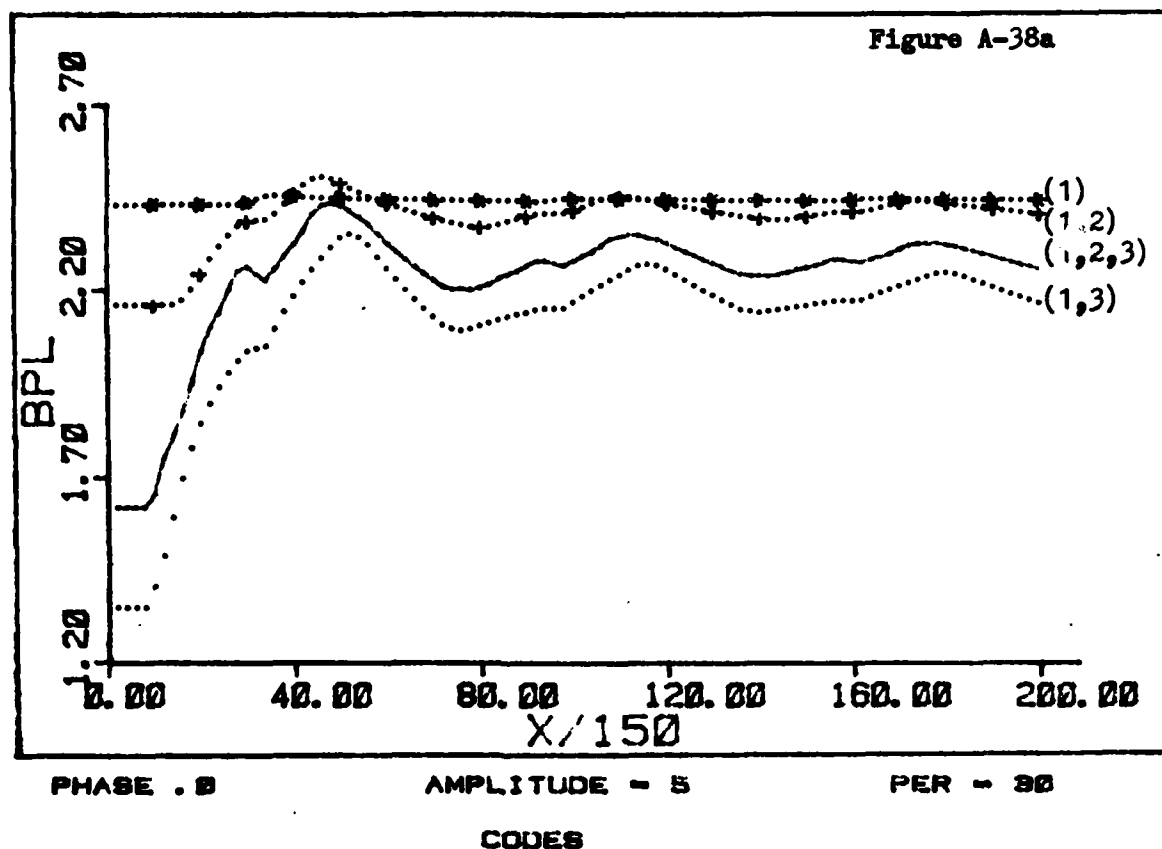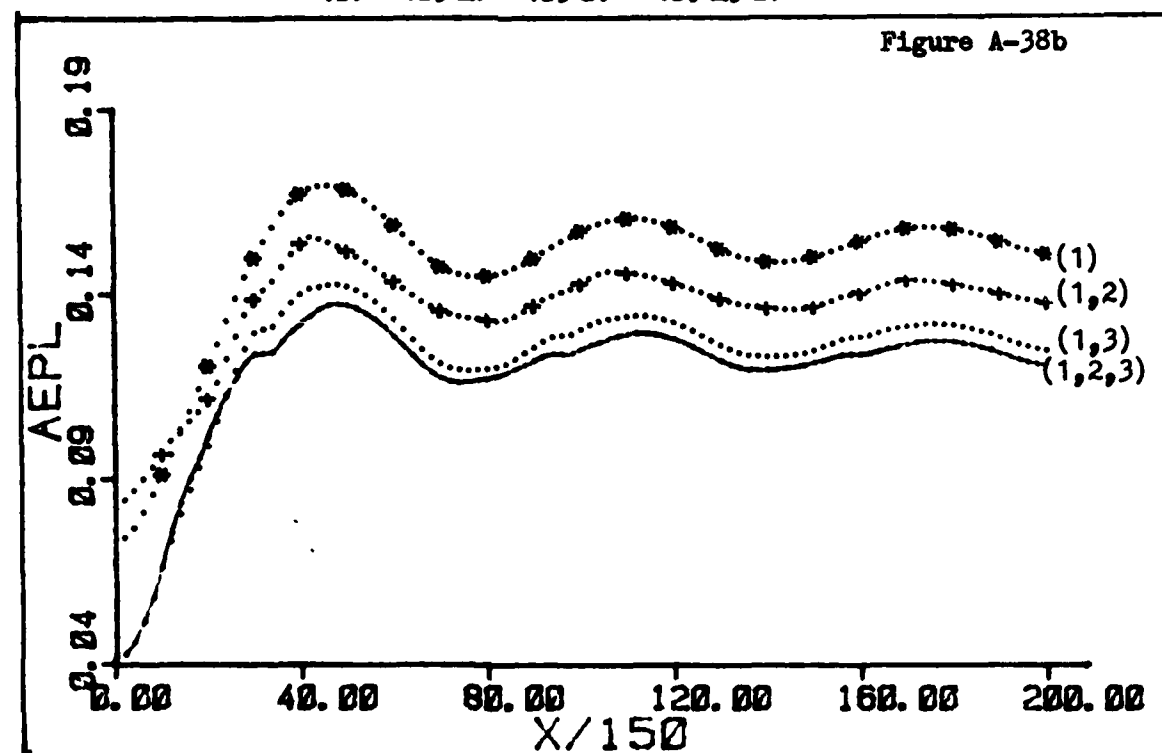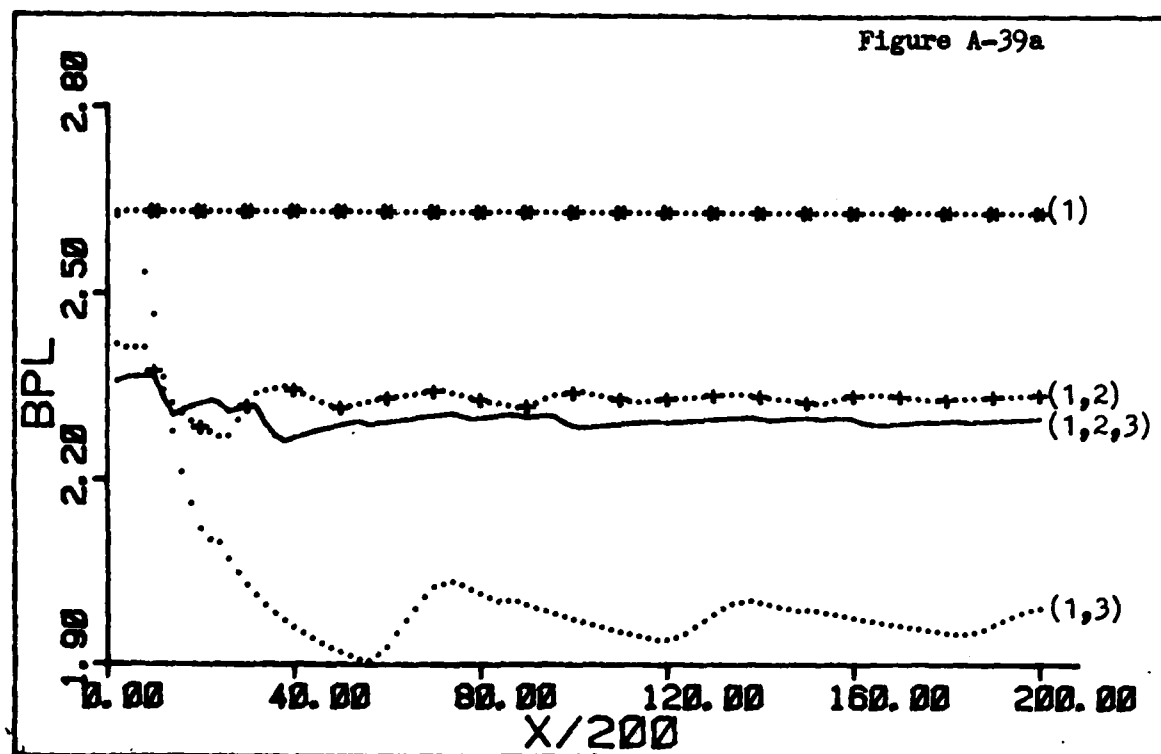
PHASE . 8   AMPLITUDE = 5   PER = 40

CODES

(1)   (1, 2)   (1, 3)   (1, 2, 3)

Figure A-39b

AEPL

X/200

A-43
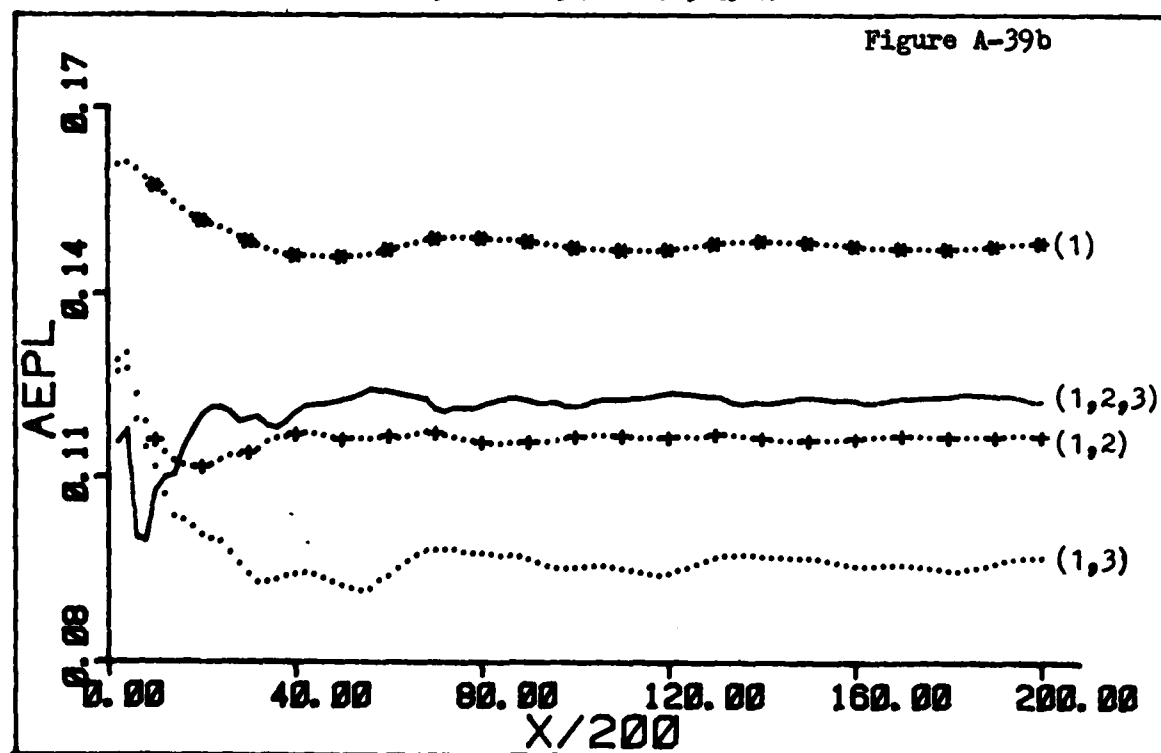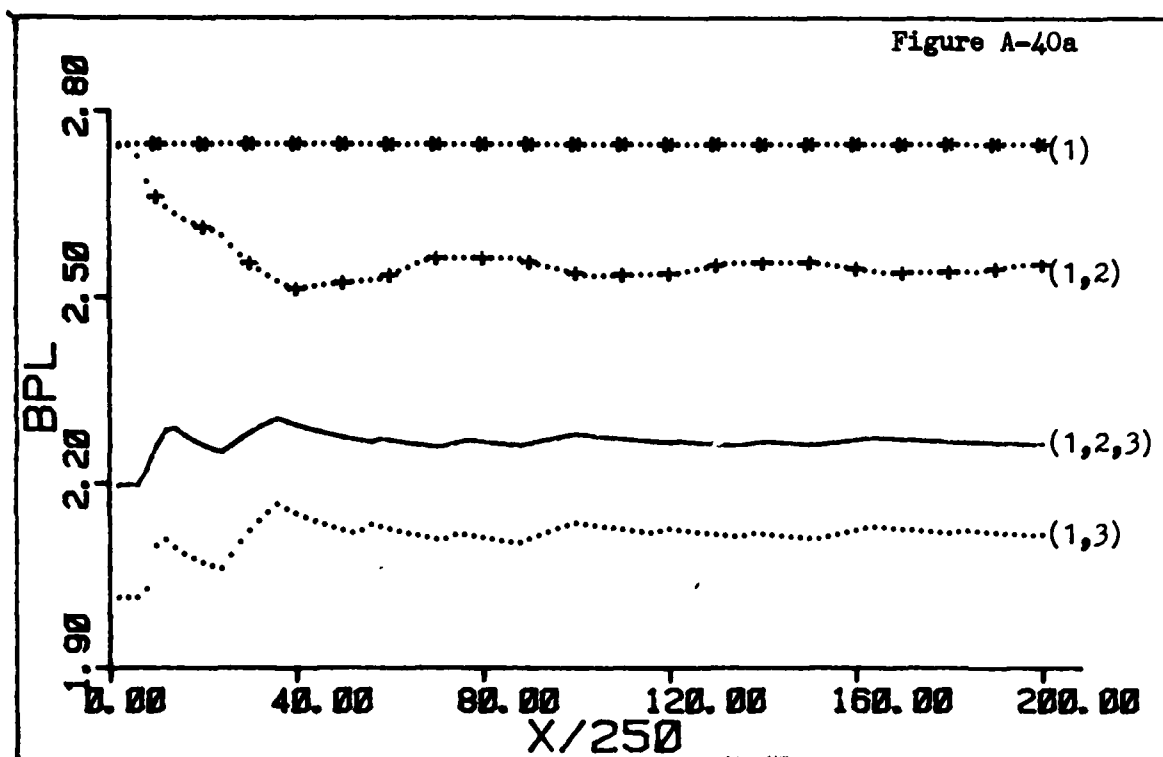
Figure A-40a

PHASE . Ø   AMPLITUDE - 5   PER - 50

CODES

(1) (1, 2) (1, 3) (1, 2, 3)

Figure A-40b

A-44
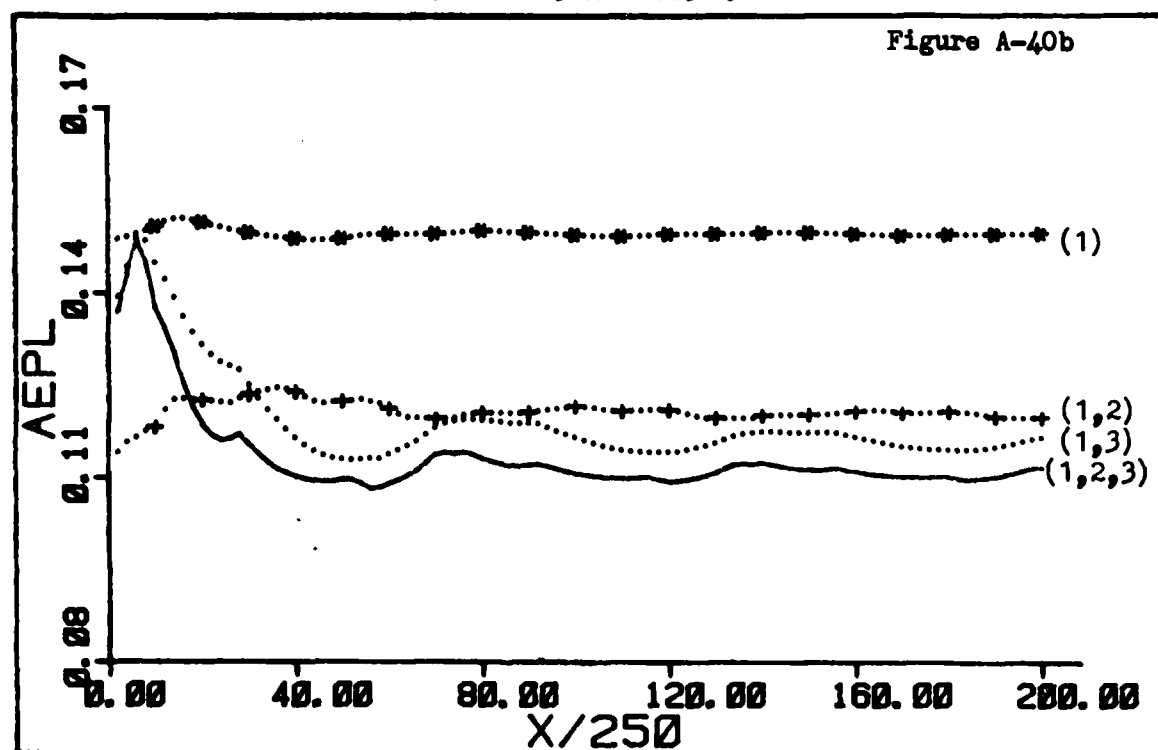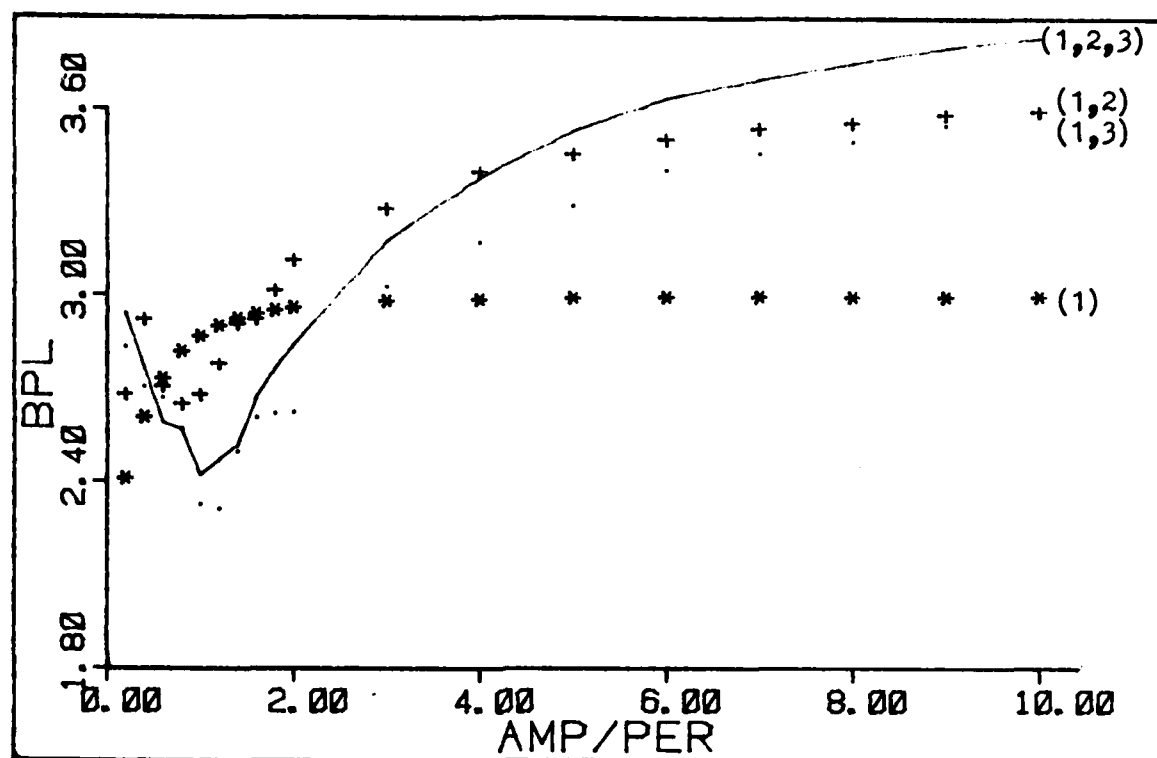
Figure A-41a

PHASE . 0   PER = 10

CODES

(1)   (1, 2)   (1, 3)   (1, 2, 3)

Figure A-41b

Figure A-42a

PHASE . 0   AMPLITUDE = 5

CODES

(1)   (1,2)   (1,3)   (1,2,3)

Figure A-42b

A-46
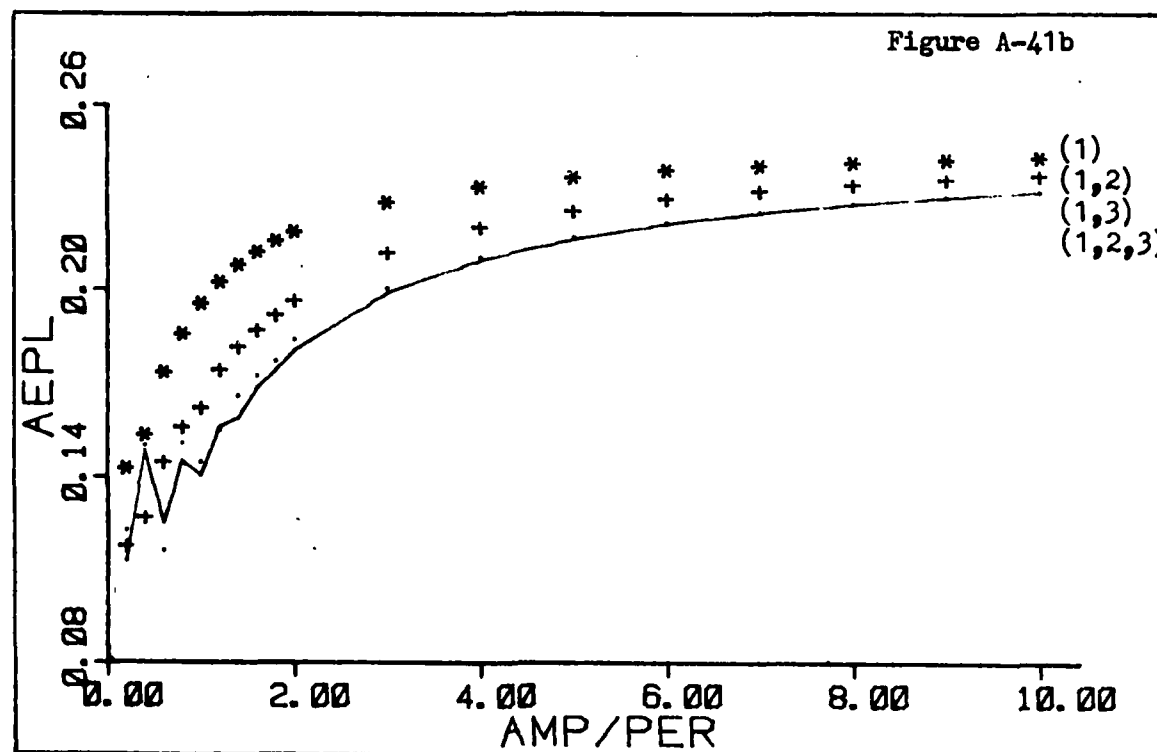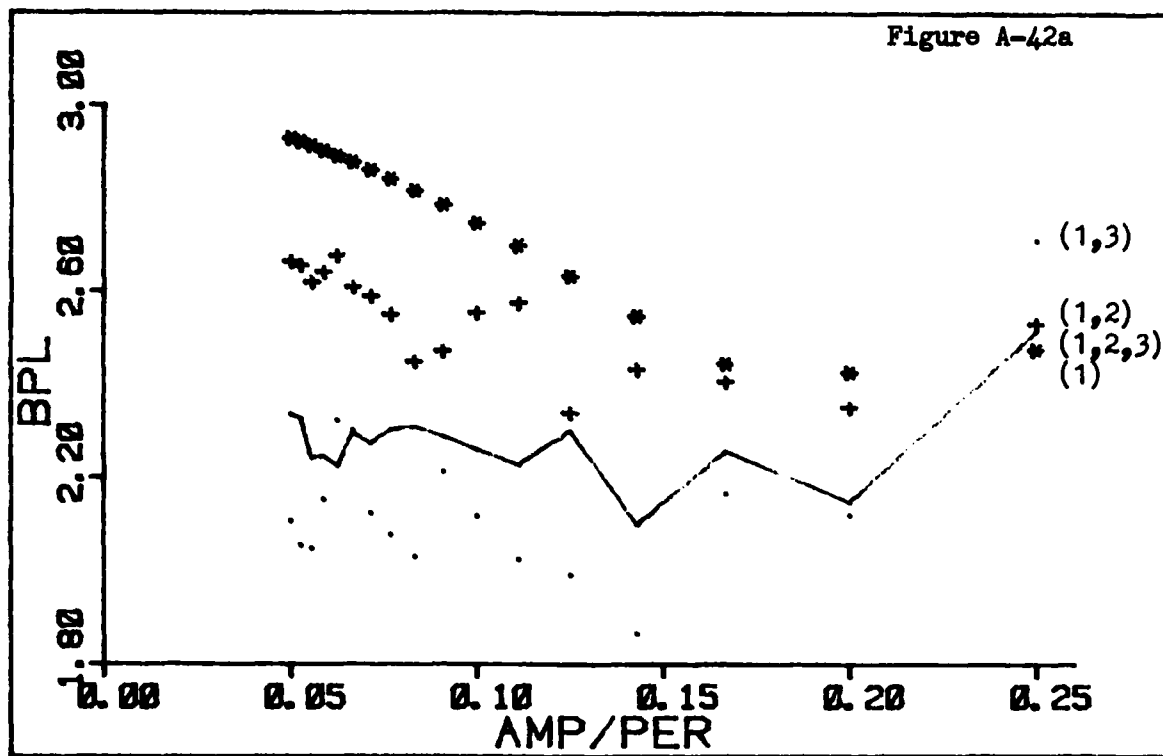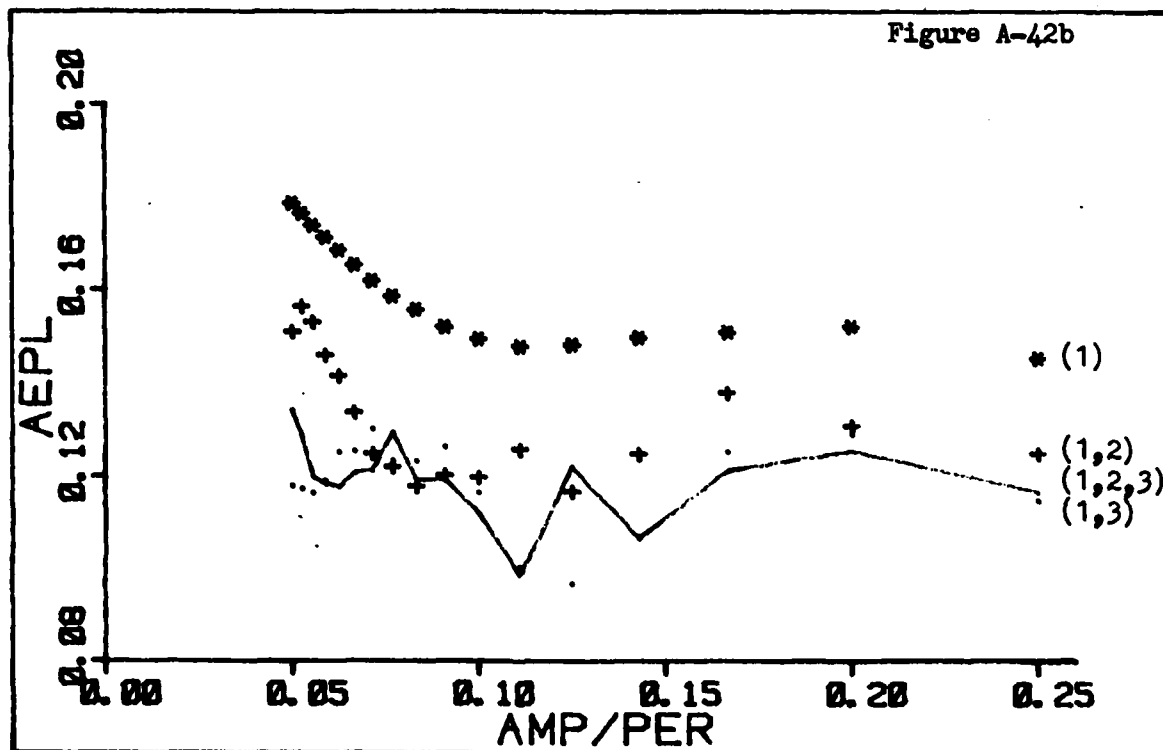
Figure A-43
Code Performance
AEPL and BPL Equally Weighted

PHASE . 0   AMPLITUDE = 5
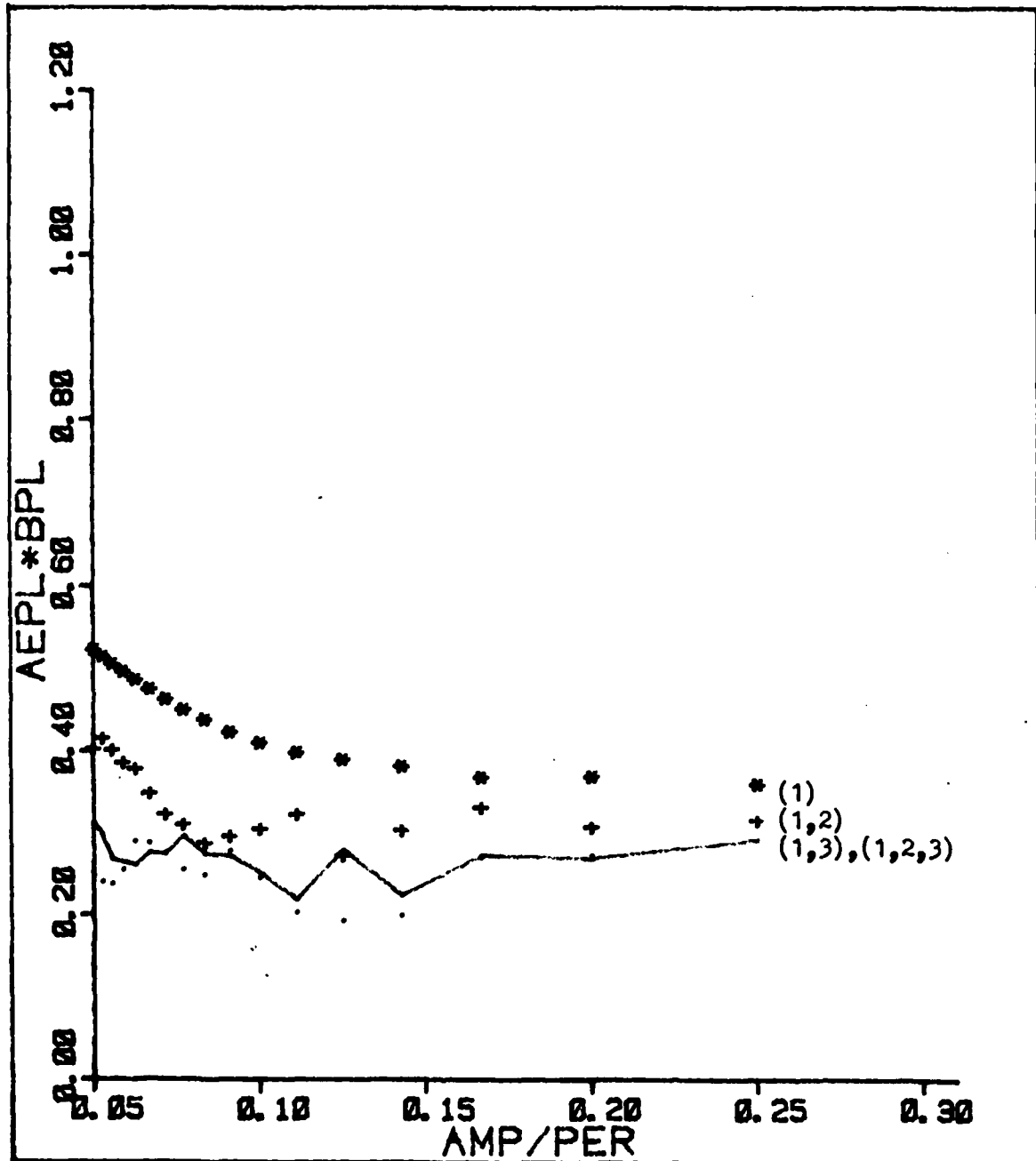
CODES

(1) (1, 2) (1, 3) (1, 2, 3)

Figure A-44
Code Performance
AEPL and BPL Equally Weighted

PHASE . Ø   PER = 1Ø

CODES

(1)   (1, 2)   (1, 3)   (1, 2, 3)

A-48
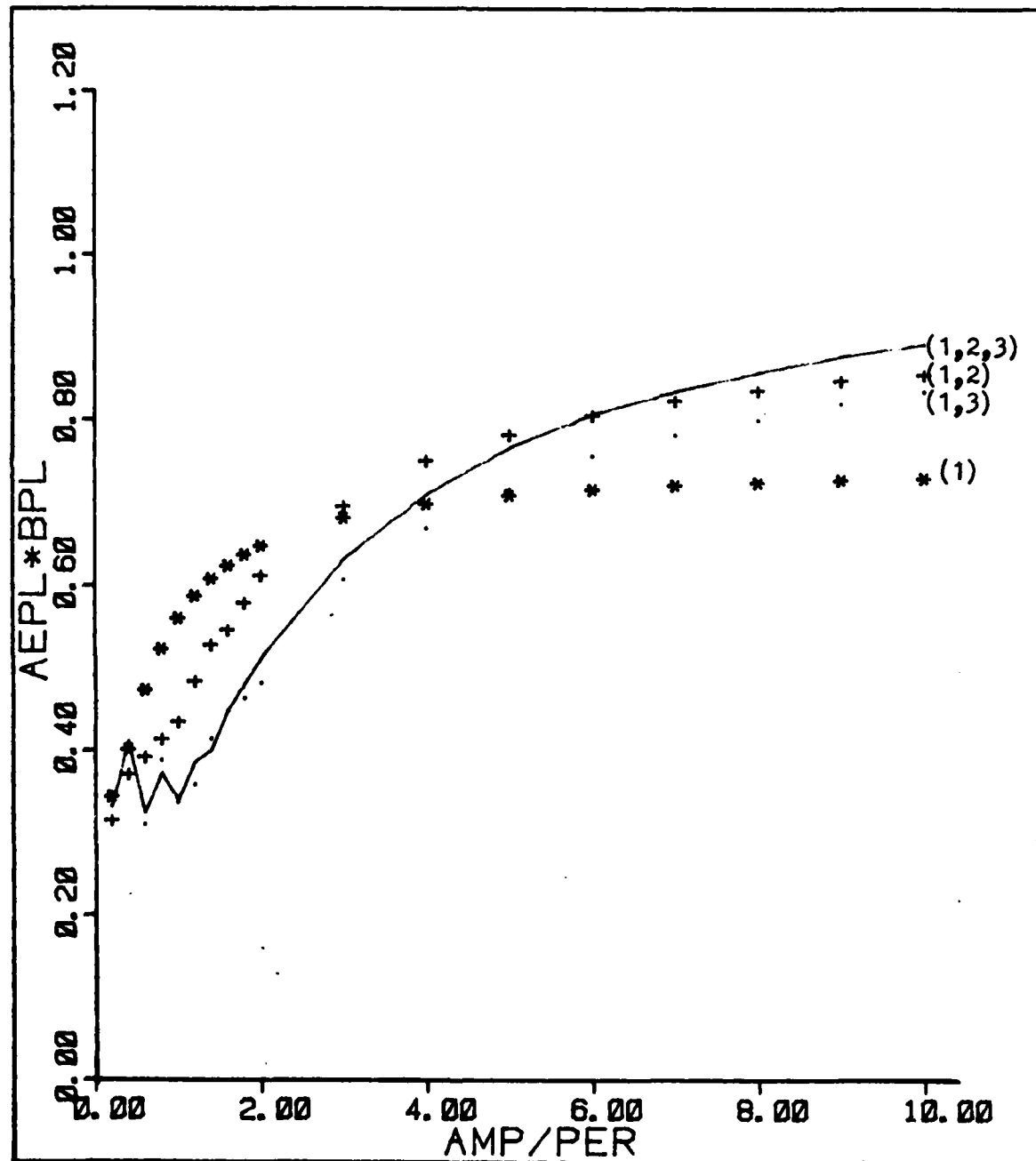
TABLE A-1

Asymptotic Values of AEPL and BPL

Code	Period	Amplitude	AEPL	BPL
1	10	500	.2482269	3.000068
1	10	1000	.2488344	3.000153
1	20	500	.2474092	2.999670
1	20	1000	.2486595	2.999976
1,2	10	500	.2470029	3.719299
1,2	10	1000	.2482250	3.734437
1,2	20	500	.2449745	2.688825
1,2	20	1000	.2474269	3.719261
1,3	10	500	.2458516	3.824547
1,3	10	1000	.2476406	3.856128
1,3	20	500	.2427406	3.760743
1,3	20	1000	.2462500	3.824410
1,2,3	10	500	.2458178	4.097472
1,2,3	10	1000	.2476272	4.131775
1,2,3	20	500	.2426481	4.030807
1,2,3	20	1000	.2462297	4.097925

TABLE A-2

Asymptotic Values of AEPL and BPL (1) Code
Period = 20

Amplitude	AEPL	BPL
10.0	.1601985	2.726612
10.1	.1612904	2.724716
10.2	.1633347	2.735440
10.3	.1670061	2.741070
10.4	.1719414	2.742734
10.5	.1779645	2.741900
10.6	.1691186	2.739371
10.7	.1682938	2.746597
10.8	.1680220	2.750871
10.9	.1666773	2.752015
11.0	.1647528	2.754504
20.0	.1957823	2.880466
20.1	.1946993	2.892659
20.2	.1957706	2.892353
20.3	.1973702	2.891657
20.4	.1993118	2.890753
20.5	.2016630	2.887158
20.6	.2002526	2.899623
20.7	.1986799	2.899171
20.8	.1975523	2.896405
20.9	.1974598	2.892388
21.0	.1979185	2.888404

TABLE A-3

Asymptotic Values of AEPL and BPL (1,3) Code
Period = 20

Amplitude	AEPL	BPL
10.0	.1103938	2.413937
10.1	.1086148	2.290830
10.2	.1113781	2.299343
10.3	.1159503	2.318190
10.4	.1212405	2.364593
10.5	.1270324	2.406615
10.6	.1125786	2.339938
10.7	.1091248	2.366146
10.8	.1087356	2.364184
10.9	.1074386	2.374788
11.0	.1054103	2.361584
20.0	.1404982	2.202335
20.1	.1380205	2.189669
20.2	.1384002	2.182735
20.3	.1400226	2.190168
20.4	.1419936	2.196358
20.5	.1440610	2.193330
20.6	.1442447	2.197877
20.7	.1425788	2.199779
20.8	.1408369	2.196193
20.9	.1403282	2.200487
21.0	.1406388	2.199150

Vita

Eric Richard Christensen was born on 30 October 1953 in North Platte, Nebraska. He graduated from high school in Maxwell, Nebraska in 1972 and was appointed to the United States Military Academy, West Point, New York. He graduated from the Military Academy in 1976 with a Bachelor of Science Degree and a Commission of Second Lieutenant in the United States Army as a Signal Corps Officer. He attended the Signal Officer Basic Course, Fort Gordon, Georgia in 1976 and then was assigned to the 52nd Signal Battalion, 160th Signal Brigade in Stuttgart, West Germany. He served as Operations Officer of the 589th Signal Company, Battalion Adjutant of the 52d Signal Battalion and then as Commander of the 589th Signal Company. He recieved a Master of Science Degree in Systems Management from the University of Southern California in January 1980. He attended the Signal Officer Advanced Course at Fort Gordon, Georgia in 1982 and then was selected to attend the School of Engineering, Air Force Institute of Technology. He was married to Miss Georgetta Gadberry on 18 June 1983.

Permanent Address: Maxwell Nebraska 69151

REPORT DOCUMENTATION PAGE	READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
AFIT/GE/EE/83D-16	AD-A138 503	

4. TITLE (and Subtitle)	5. TYPE OF REPORT & PERIOD COVERED
GRID-BASED LINE DRAWING QUANTIZATION	MS THESIS
	6. PERFORMING ORG. REPORT NUMBER

7. AUTHOR(s)	8. CONTRACT OR GRANT NUMBER(s)
Eric R. Christensen CPT, USA	

9. PERFORMING ORGANIZATION NAME AND ADDRESS	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
Air Force Institute of Technology (AFIT-EN) Wright-Patterson AFB, Ohio 45433	

11. CONTROLLING OFFICE NAME AND ADDRESS	12. REPORT DATE
	December 1983
	13. NUMBER OF PAGES
	128

14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)	15. SECURITY CLASS. (of this report)
	Unclassified
	15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

Approved for public release: IAW AFR 190-17.

Dean for Research and Professional Development
Air Force Institute of Technology (AIC)
Wright-Patterson AFB OH 45433

7 Feb 84

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Line Drawing Quantization
Generalized Chain Codes
Triangular Quantization Scheme
Parallel Quantization Scheme

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This paper documents a quantitative analysis of the performance of the generalized chain codes when used to quantize sinusoids with specific periods and amplitudes. The analysis was performed using a software simulation of the various generalized chain codes using the triangular quantization scheme. The performance of the codes was measured in terms of encoding rate and the area error in the quantization.

DD , FORM 1473 EDITION OF 1 NOV 68 IS OBSOLETE
1 JAN 73

Block 20 contued

Comparisons were made of the performance of the codes as the amplitude to period ratio was changed. Comparisions were made when the same codes were used to quantize the same sinusoids. An attempt was made to quantize a rotated sinusoid, however, the algorithm implemented was too inefficient with respect to computer processing time. Finally the effect of a capture region of 1/2 of the grid size on the performance of the code was examined.

END

FILMED

3-84

DTIC